

AD-A068 172

FEDERAL AVIATION ADMINISTRATION WASHINGTON DC OFFICE --ETC F/G 1/5
ENVIRONMENTAL DATA BANK. VOLUME II. SYSTEMS MANUAL.(U)
MAR 79

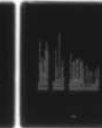
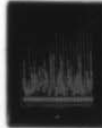
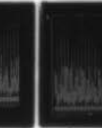
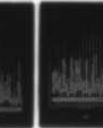
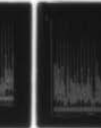
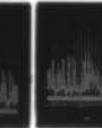
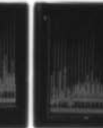
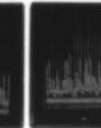
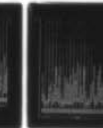
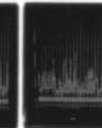
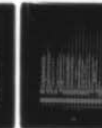
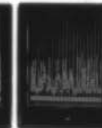
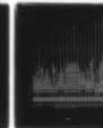
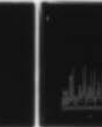
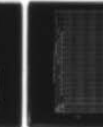
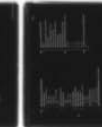
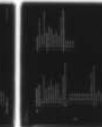
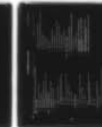
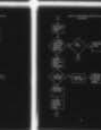
UNCLASSIFIED

FAA-AEE-79-01-V01 -2

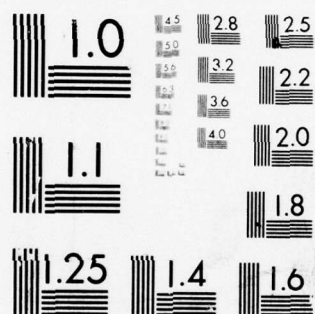
NL

1 of 1

AD
A068172



END
DATE
FILMED
6 --79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL

EDB

12
B.G.

AD A068172

DDC FILE COPY

ENVIRONMENTAL DATA BANK



VOLUME II
SYSTEMS MANUAL

MARCH 1979

U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
Office of Environment and Energy
Washington, D.C. 20591

79 04 25 004

NOTICE

The data in the EDB was compiled through January 1979. It is recommended that the EDB be used as a general reference tool only. Information contained in the EDB should be verified when using in a significant manner.

Technical Report Documentation Page

1. Report No. FAA-EE-79-01-Vol II-2		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Systems Manual for the Environmental Data Bank. Volume II. Systems Manual		5. Report Date March 1979		6. Performing Organization Code	
7. Author(s) Trans Systems Corporation		8. Performing Organization Report No. 1259p		9. Work Unit No. (TRAIS)	
9. Performing Organization Name and Address Trans Systems Corporation Madison Building 118 Park Street, S. E. Vienna, Virginia 22180		10. Contract or Grant No.		11. Type of Report and Period Covered Systems Manual	
12. Sponsoring Agency Name and Address Department of Transportation Federal Aviation Administration Office of Environment and Energy Washington, D.C. 20591		13. Sponsoring Agency Code		14. Supplementary Notes	
15. Abstract See the following Abstract page.					
17. Key Words Airport-Specific Environmentally-Oriented Data Land Acquisition Programs Noise Control Actions			18. Distribution Statement Document is available to the public through the National Technical Information Service Springfield, Virginia 22151		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 61	
				22. Price	

411031

JP

B

ABSTRACT

Documentation of the Environmental Data Bases

✓ The Environmental Data Bank (EDB) represents an effort to compile a comprehensive listing of environmentally-oriented data within one convenient source. The data were collected with the aid of the FAA Regional Offices and include airport-specific information regarding the existence of such things as land acquisition programs or other such noise control actions in effect at each of the U.S. airports listed as of March 1979. FAA Form 1050-5 (Volumes I and II, Appendix A) summarizes those kinds of data which may be listed (if applicable) at each airport.

The individual airport data themselves are arranged in the alphabetical order of the airport's "location indicator" (LOCID), by FAA region. These data may not reflect all U.S. airports having significant environmental information, as we have only indicated those data here that have been reported to us. This EDB may be useful by providing a sense of the extent to which environmentally-related activities have affected approximately 475 of our Nation's airports.

Briefly, the EDB is organized into four individual volumes:

Volume I - User's Manual gives detailed information about the content and use of the data base.

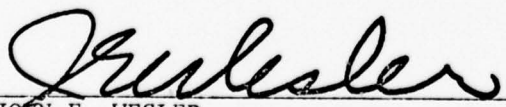
→ Volume II, - Systems Manual, contains a description of the system and programs that support the use and management of the data within the file. ✕

Volume III- Airport Environmental Data Manual presents site-specific information for each airport included in the data base.

Volume IV - Airport Supplemental Information Manual contains supplemental data of special environmental conditions and/or problems (not provided for on Form 1050-5) for each airport included in the data base.

Consistent with the format and use of the U.S. airports' EDB, an International Environmental Data Bank (IEDB) was developed through information provided by the International Civil Aviation Organization (ICAO). This international data, as documented in the IEDB volume, is a subset of the larger environmental data file and is available separately. The IEDB volume contains airport-specific information for approximately 110 foreign airports.

It is intended that subsequent updates of all of these documents will be produced as required by changing circumstances.


JOHN E. WESLER
Acting Director of Environment and Energy

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DOC	<input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFIED	<input type="checkbox"/>
BY	
DISTRICT	
D	
SIAL	
A	

TABLE OF CONTENTS

<u>Chapter</u>	<u>Page</u>
1.0 INTRODUCTION	1
2.0 SYSTEM SPECIFICATION AND CONCEPTS	2
3.0 SYSTEM OVERVIEW	3
3.1 Application Description	3
3.2 The Environmental Data Bank (EDB)	3
3.3 System Description	4
3.4 Data Input Forms and Data Bank File Structures	6
3.5 Creation of the Data Bank and Data Entry Procedures	6
3.6 Data Bank Update	6
4.0 GENERAL PROCEDURES FOR SYSTEM OPERATION	7
4.1 Terminal Procedure	7
5.0 PROGRAM SPECIFICATIONS AND DESCRIPTIONS	9
5.1 Procedure	9
5.2 Program USE8	10
5.3 Program USE5	15
5.4 Program UPDATE	15

APPENDICES

	<u>Page</u>
Appendix A FAA EDB Form 1050-5 and Coding Form	A-1
Appendix B Program Listings	B-1

ILLUSTRATIONS

<u>Figure</u>		
3 -1	EDB System Overview	5
5-1	Program USE8 Flowchart	11
5 -2	Program USE8 CLIST Flowchart	14
5 -3	EDB Program UPDATE Flowchart	16
5-4	Program UPDATE CLIST Listing	21
5-5	Sample Computer Output of the USE8	22
5-6	Sample UPDATE Program Execution and Outputs	23

CHAPTER 1

INTRODUCTION

The System Manual provides detailed information and technical description necessary for data processing personnel to understand, use, and maintain the EDB computer program. The manual gives a full description of the EDB System, program descriptions, the procedure for data entry into the system, and the update and maintenance procedures.

The Manual also covers the running procedure and instructions for producing a report. This system has been written in a clear, simple manner; and all information and descriptions are given in such a way that the system can be fully used, and all of its capabilities fully utilized so that the required results can be obtained.

For information regarding access to the system by the user and the production of reports, please refer to Volume I of the Manual - the User's Manual.

CHAPTER 2

SYSTEM SPECIFICATIONS AND CONCEPTS

The EDB System will be operated on any IBM 360/370 OS computer system that will support the time-sharing option (TSO). COBOL has been used as the programming language, and standard OS utilities, such as SORT, have been used.

The EDB currently operates at the Boeing Computer Service (BCS) on an IBM 370/168 computer, utilizing the BCS MAINSTREAM TSO, which is a BCS-enhanced version of the IBM OS/VS2 Multiple Virtual Storage (MVS) System. MAINSTREAM TSO is upwardly compatible with the IBM TSO system under OS/VS2 MVS.

TSO was selected because it allows a user to interface with a computer system easily, and allows the user to utilize fully the capabilities of the computer through the use of a remote terminal located many miles away.

Within the EDB, TSO provides a mechanism for:

- o The input of raw airport-use data.
- o The execution of the various components of EDB.
- o The generation of reports.
- o The creation and modification of computer programs for use within the EDB system.

Use of the EDB does not require a great deal of knowledge about TSO. One need only know how to LOGON to the computer (see section on System Access in the User's Manual) and how to request the execution of necessary programs (see section on procedures for execution of each of the programs).

The interface with the computer by the user has been further simplified through the use of various CLISTs under TSO. A CLIST is a command procedure that provides a convenient means of executing a frequently used sequence of commands. By incorporating some of the functions of a computer programming language, a CLIST allows a user to access the system, manipulate data, and process jobs, without knowing a programming language.

CHAPTER 3

SYSTEM OVERVIEW

3.1 Application Description

The EDB system's purpose is to allow users to access the environmental data file for the public airports and to produce reports as desired from the data file.

The information in the Data Bank is a result of compilation of nationwide data relating to airport oriented actions taken to control, reduce, and/or minimize the impact of aircraft operations. Various items regarding state and local government actions, and local activities and conditions, are recorded for each of the approximately 500 major civil airports in the United States.

The facilities of the EDB system allow the user to create a Data Bank from raw airport-use data, to maintain the Data Bank by adding new data or modifying/deleting existing data, and to produce the necessary reports, either in an interactive or batch mode. Small reports, such as information on specific airports, may be produced interactively. For the larger or the full reports, the batch capabilities of the system should be utilized.

3.2 The Environmental Data Bank (EDB)

The Data Bank consists of two basic files. The first file consists of EDB data for all the airports in a sorted form; its data set name is FAA130. EDB. AUG78. USE8. DATA. The second file is the decode file, containing the textual equivalence for all airport location and restriction-use codes; its data set name is FAA110. MAST. DECODE. AUG78. DATA.

In addition, there are other files that serve as program files, backup files, and temporary files.

All of the files and their associated data set names are given below:

- o FAA130. EDB. AUG78. USE8. DATA
(This file contains the EDB data for all airports in sorted form.)
- o FAA119. DECODE. AUG78. UNLOADED. DATA
(This files contains the textual equivalence for all airport LOCID codes and restriction-use codes.)

- o FAA112. MAST. DECODE. CNTL(CREATE)
(This file contains job control statements to reload the Decode file.)
- o FAA110. MAST. DECODE. AUG 78. DATA
(This is the decode file.)
- o FAA130. EDB. COBOL(USE8)
(This is a COBOL program which produces a printed report and an EASYTRIEVE query file.)
- o FAA118. AIRPORT. MAR 78. USE. EASYTREV. DATA
(This is an EASYTRIEVE file generated by the COBOL(USE8) program.)
- o FAA130. EDB. COBOL(USE5)
(This is a COBOL program that produces a printed report.)
- o FAA130. EDB. LOAD(UPDATE)
(This is a COBOL program that is used to add records to the data file or to update existing records.)

The relationships between these files are shown in the system flowchart included in Section 3. 3, 'System Description. '

3. 3 System Description

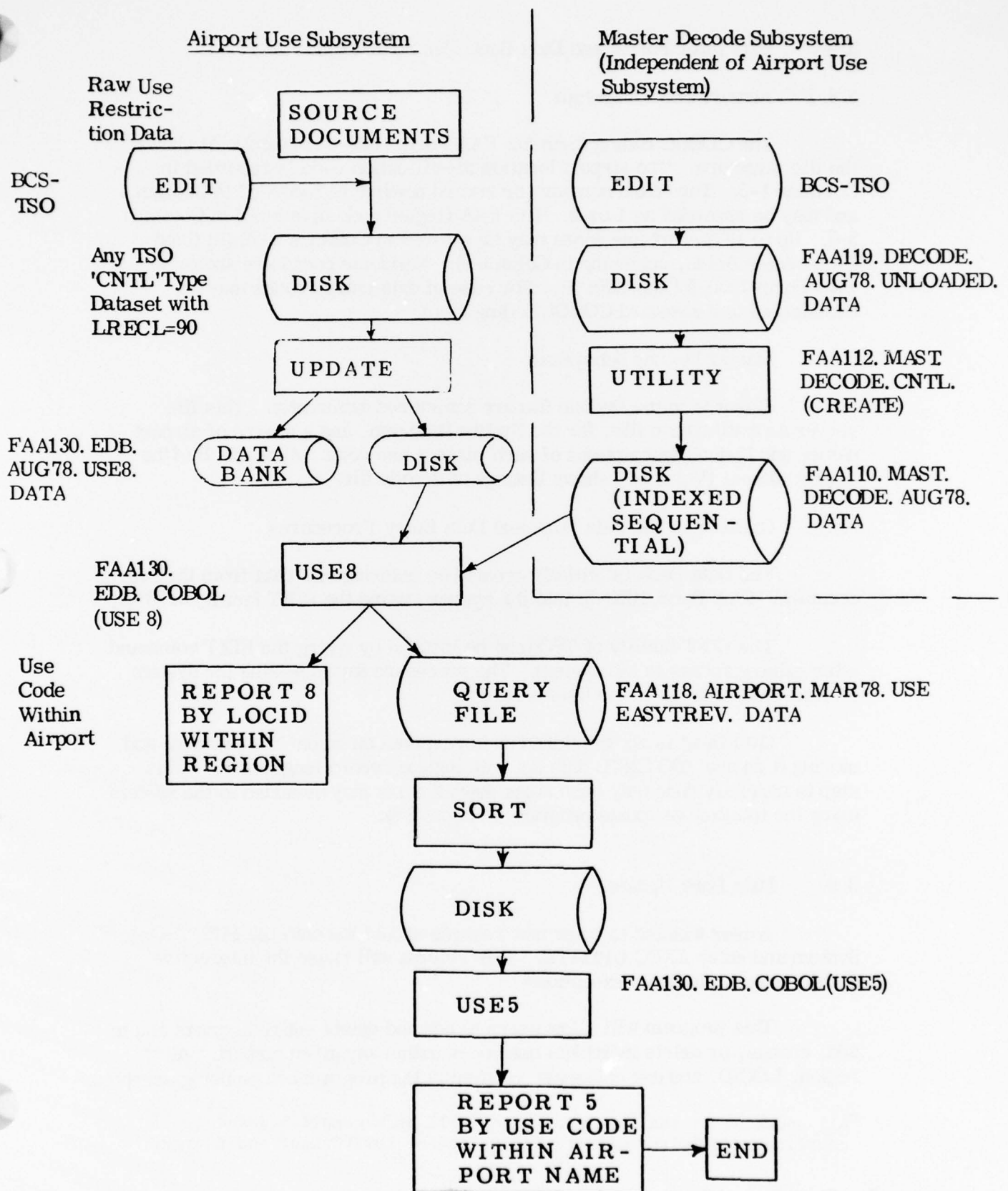
The system flowchart shown in Figure 3-1 provides an overview description of the system. The EDB System consists of the Airport Use Subsystem and the Master Decode Subsystem.

The Airport Use Subsystem provides for the capability of creating and maintaining the Data Bank, and for producing the desired reports. Data from source documents are keyed into the system under the EDIT facility of TSO. Data are stored on DISK and can then be loaded into the EDB Data Bank by using the interactive update program. Additional data input and changes/deletions to the Data Bank can be performed by the Update Program.

Program USE8 provides the means of producing the various reports, which are discussed in subsequent sections.

The Master Decode System provides for the capability of creating and maintaining the decode file. The file contains the textual information for each of the restriction codes in the Data Bank.

Figure 3-1. EDB System Overview



3.4 Data Input Forms and Data Bank File Structures

3.4.1 Airport Use Subsystem

The COBOL coding form for FAA Form 1050-5 (Appendix A) shows the file structure. The airport location identification code is recorded in Columns 1-3. The "card number" or record number is recorded in Column 4 and may be recorded as 1 or 2. The FAA Region code is entered in Columns 5-7. Up to 18 airport use codes may be entered in Columns 8-79 (in fixed 4-character fields, beginning in Column 8). Valid use codes are shown on FAA Form 1050-5 (Appendix A). For ease of data entry, codes may be transferred to a standard COBOL coding form.

3.4.2 Master Decode Subsystem

Records in the Decode file are structured differently. This file serves as a validation file, for the Update Program, and a source of airport names and textual descriptions of each airport use code. Appendix B of the User's Manual (Volume I) shows listings of decode file.

3.5 Creation of the Data Bank and Data Entry Procedures

The Data Bank is initially created by entering raw data from the source document (FAA Form 1050-5) into the system, using the EDIT facility of TSO.

The EDIT facility of TSO may be invoked by typing the EDIT command after gaining access to the system. The procedure for accessing the system is described in detail in the User's Manual.

Data input is accomplished by keying data on an on-line terminal and storing it on any TSO CNTL data set with logical record length of 90. This step is normally done only once since new airports may be added to the system using the interactive update program (Section 3.6).

3.6 Data Bank Update

A user wishing to enter new records should log onto the BCS-TSO System and enter EXEC UPDATE. This request will cause the interactive update program to begin execution.*

This program will allow users to add and delete entire airports and to add, change, or delete individual use codes within any given airport. All region, LOCID, and use codes are verified by the program before being accepted.

*All updating of the EDB information will be accomplished and coordinated only through the FAA Office of Environment and Energy.

CHAPTER 4

GENERAL PROCEDURES FOR SYSTEM OPERATION

The general procedures for maintaining the programs are described below and may be used to maintain the programs discussed in subsequent sections. The procedures allow for modifying programs through the EDIT facility of TSO and for testing the programs through the Interactive facilities. File and program listings can be obtained via the EDIT facility.

4.1 Terminal Procedure

The following instructions and procedures are for using the terminal through the TSO system:

4.1.1 Logging On

1. Connect the terminal to the computer service by dialing the assigned number through the telephone.
2. Hit Carriage Control; a prompt will be received. Select desired service (CTS, CTS2, or TSO).
3. Type TSO and hit Carriage Control.
4. A line print of MAINSTREAM-TSO will appear.
5. Type LOGON, hit Carriage Control.
6. A statement asking for the user's ID will appear.
7. Type assigned ID and hit Carriage Control.
8. An instruction comes back to type the password.
9. Type the password authorized, hit Carriage Control.
10. The instruction that follows asks for the account number.
11. Type the assigned account number, then hit Carriage Control.
12. When READY appears, enter the appropriate commands for accomplishing the operation.

4.1.2 Logging Off

When the work is completed for a session, simply logoff by keying **LOGOFF**.

CHAPTER 5

PROGRAM SPECIFICATIONS AND DESCRIPTIONS

Three major programs and the procedures for structuring additional collection of airport data are described in this section. The USE8 program is a report generator that produces sorted listings of the Data Bank. The USE8 program will also generate a query file if executed in batch mode only. Finally, the Update program allows the user to add, delete, and change records as desired.

Each program may be executed with the EXEC command of TSO. The program will then be loaded and the user will be prompted for run instructions. In the case of USE8, the user may elect to run in batch or interactive mode. The Update program is exclusively interactive. Appendix B contains the four COBOL program listings.

The airport-use data was collected following the creation of the original Bank. The new data are merged with the original EDB data. Once all of the data are keyed, the data are loaded into the EDB System. When this has been done, the loading need not be repeated unless an entirely new survey is undertaken because of updating capability.

5.1 Procedure for Program Operation

1. Log on TSO (described in 4. 1).
2. When computer responds READY, enter 'EDIT 'FAA130.EDB. MMYYY. DATA' DATA LRECL(90)' and hit carriage return (CR). The 'MMYY' stands for the month (JAN-DEC) and year (78-99). of the survey.
3. The computer will respond with a line number. Then begin to enter additional survey data in the following format:
 - (a) Columns 1-3 LOCID
 - (b) Column 4 Coding line number (1 or 2)
 - (c) Columns 5-7 Region
 - (d) Columns 8-79 Airport use codes (18 maximum), 4 characters each.

4. After all data have been entered, then respond with a blank line (CR) without entering any data. This will end the input phase and return to the edit mode. Then enter SAVE and END. Now the data are saved permanently. To enter additional survey data into the EDB, merely repeat the previous instructions. For more information on TSO or EDIT, consult the appropriate IBM publications. Additional data entry through the Update program is described in Section 5. 4.

5. 2 Program USE8 (EDB Print Program)

5. 2. 1 Abstract

The USE8 program generates a report which lists the FAA region, airport location identification code, airport name, and each airport use code and textual description of those codes for each airport in the Data Bank. The report, when generated from a sorted file, provides all restriction data for each airport by FAA region. USE8 (when run in a BATCH mode) also produces a copy of the EDB Data Bank in an EASYTRIEVE format. Figure 5-1 shows the Program USE8 flowchart.

5. 2. 2 Description

5. 2. 2. 1 Input

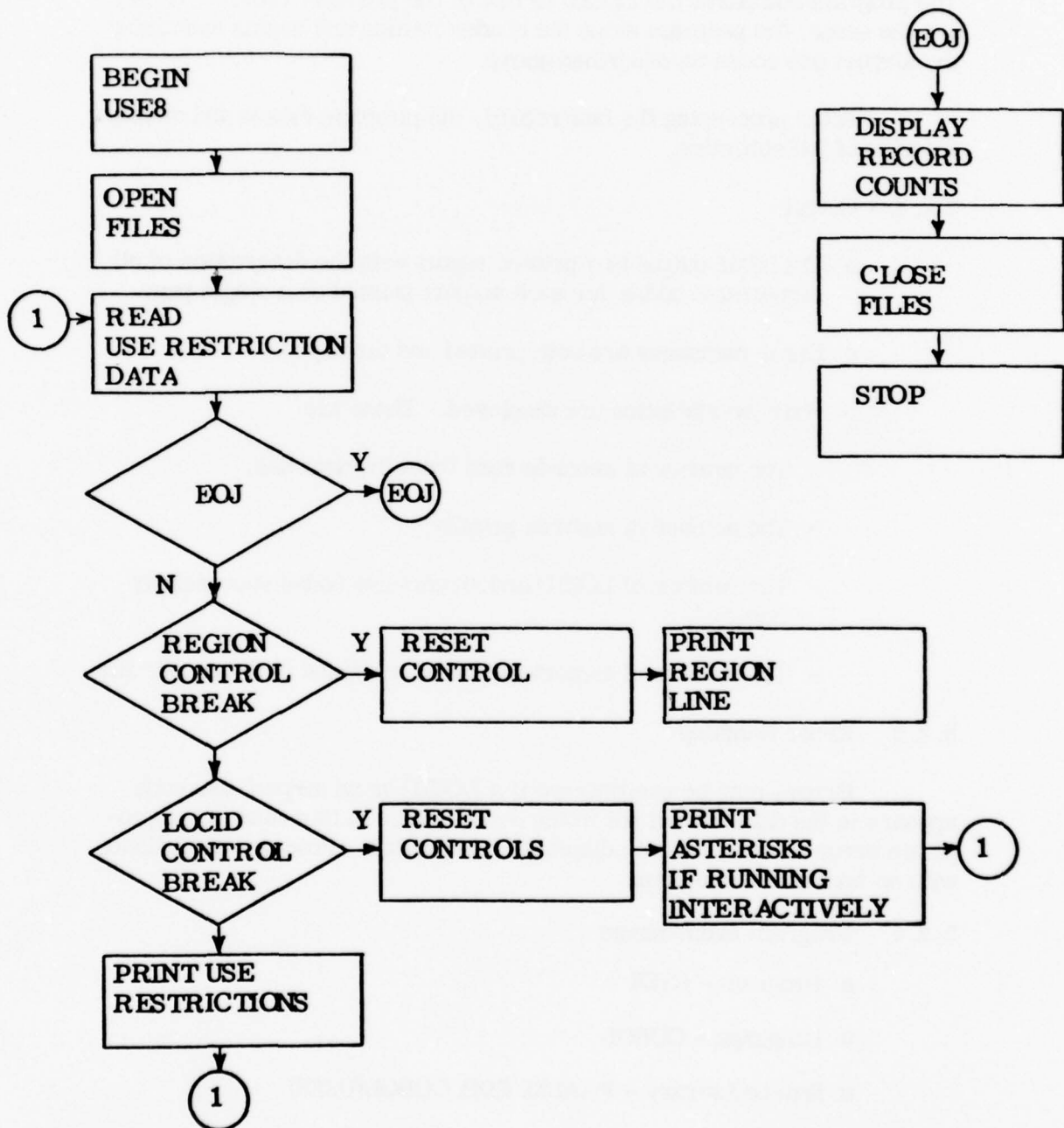
- o Airport use data file: DSN=FAA130. EDB. AUG78. USE8. DATA
(sorted by region and LOCID)
- o Master decode file: DSN=FAA110. MAST. DECODE. AUG78. DATA

5. 2. 2. 2 Processing

The USE8 program reads each record from the USE8 DATA file sequentially. The program searches the master data and decode file for the same LOCID to determine the name of the airport. If the LOCID is not found in the decode file, an error message is both displayed and printed. If the LOCID is found in the decode file, the program prints a header line giving the FAA Region code, LOCID, and airport name.

After printing a valid header, or an error message, the program compares each airport-use code (maximum of 18 per record) with the decode file to determine the textual equivalence of each code. If an airport-use code is not found in the decode file, an error message is both displayed and printed,

Figure 5-1. Program USE8 Flowchart



and the program moves to the next restriction code. If the airport use code is found, the program prints both the code and its textual equivalence, then moves to the next restriction code. When all airport use codes in a record have been examined, the program moves to the next record.

It is possible for a single airport to have more than one record in the data file (i. e., to have more than 18 airport-use codes attached to it). The program compares the LOCID to that of the previous record. If they are the same, the program skips the header routine and begins examining the airport use codes as described above.

After processing the last record, the program closes and displays a series of job statistics.

5. 2. 2. 3 Output

- o The basic output is a printed report with the description of all airport-use codes for each airport printed on a single page.
- o Error messages are both printed and displayed.
- o Four job statistics are displayed. These are:
 - The number of records read from the data file,
 - The number of records printed,
 - The number of LOCID and airport-use codes successfully decoded,
 - The number of airport-use codes not found in the decode file.

5. 2. 3 Error Handling

Errors may be encountered if a LOCID or an airport-use code appears in the data file but not in the decode file. In this case, an appropriate error message is both displayed and printed. Processing continues with no further interruption.

5. 2. 4 Program Maintenance

- o Program - USE8
- o Language - COBOL
- o Source Library - FAA130. EDB. COBOL(USE8)

5. 2. 4. 1 Modifying the Program

To make modifications to this program you should EDIT the source library data set on BCS-TSO, make appropriate changes, and resave the program into the source library. Please refer to the section on general procedures for program maintenance for detailed procedures.

NOTE: If extensive modifications are required, it is recommended that the user make a user copy of the program, modify it, test and debug it, and then resave.

5. 2. 4. 2 Recompiling the Program

- o LOGON BCS-TSO
- o EXEC COBOL
- o Enter data set name - EDB.COBOL(USE8)

The program will be read from FAA130. EDB.COBOL., compiled and linkage-edited into FAA130. EDB.LOAD.

5. 2. 5 Command Procedures for Producing Reports from Program USE8

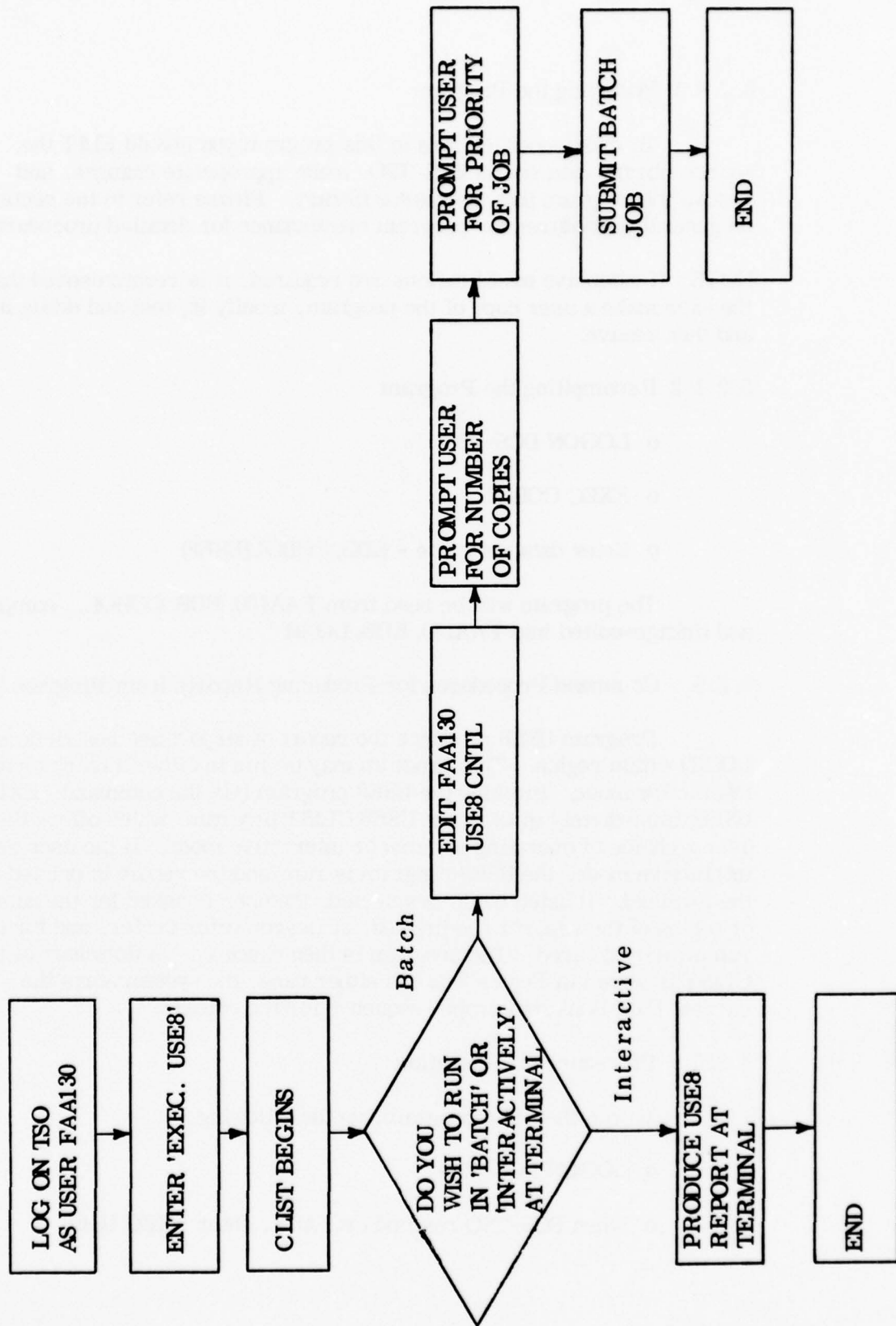
Program USE8 produces the report of airport use restrictions by LOCID within region. This program may be run in either a batch or an interactive mode. Invoking the USE8 program (via the command: EXEC USE8) immediately invokes the USE8 CLIST program, which offers the user a choice of operating in batch or interactive mode. If the user selects interactive mode, the USE8 program is run, and the report is printed at the terminal. If batch mode is selected, the user is asked for the number of copies of the report to be printed, at the computer center, and for the run priority desired. The program is then executed. A flowchart of the CLIST is shown in Figure 5-2. In either case, the system sorts the current Data Bank into proper sequence for the report.

5. 2. 6 Procedure for Execution

To run the USE8 program, do the following:

- o LOGON BCS-TSO
- o When BCS-TSO responds READY, enter EXEC USE8.

Figure 5-2. Program USE8 CLIST Flowchart



User will then be prompted to see if BATCH or INTERACTIVE execution is desired. If INTERACTIVE is chosen, the program will begin to execute at the terminal. If BATCH is chosen, the user will be prompted for the number of copies needed and the priority of the run.

Remember: THE HIGHER THE PRIORITY, THE FASTER THE JOB RUNS,
AND THE MORE IT COSTS.

Pickup must be arranged by the user of this listing by Boeing Computer Services, or their delivery services may be utilized if run in a batch mode.

5.3 Program USE5 (Print Report)

USE5 program exists in source-code format within the EDB. COBOL library only. The usefulness of this report program has been reduced significantly with the USE8 program and the interactive update program.

5.4 Program UPDATE

5.4.1 Abstract

The program UPDATE is used to add new records to the data file (i. e., to add a previously unlisted airport and its restriction data) or to change airport-use data in existing records (i. e., to add, delete, or change one or more restriction codes from a record). This is an interactive program under TSO. Figure 5-3 shows flowcharts of the EDB Update Program.

5.4.2 Description

5.4.2.1 Input

- o Airport use file update: DSN=FAA130. EDB. DATABASE
(created by the program from the raw survey data keyed into the computer.)
- o Decode data file: DSN=FAA110. MAST. DECODE. AUG 78.

5.4.2.2 Processing

After signing onto the system and invoking the Update program, the system will respond with a series of commands designed to locate the record that the user wishes to change. First, the program asks for a FAA region code or a command (such as LIST). If a command is entered, it is processed, and the program asks for another region. If an invalid region (or command) is entered, the program indicates an invalid response and asks the user to re-enter the region or command.

Figure 5-3. EDB Program UPDATE Flowchart

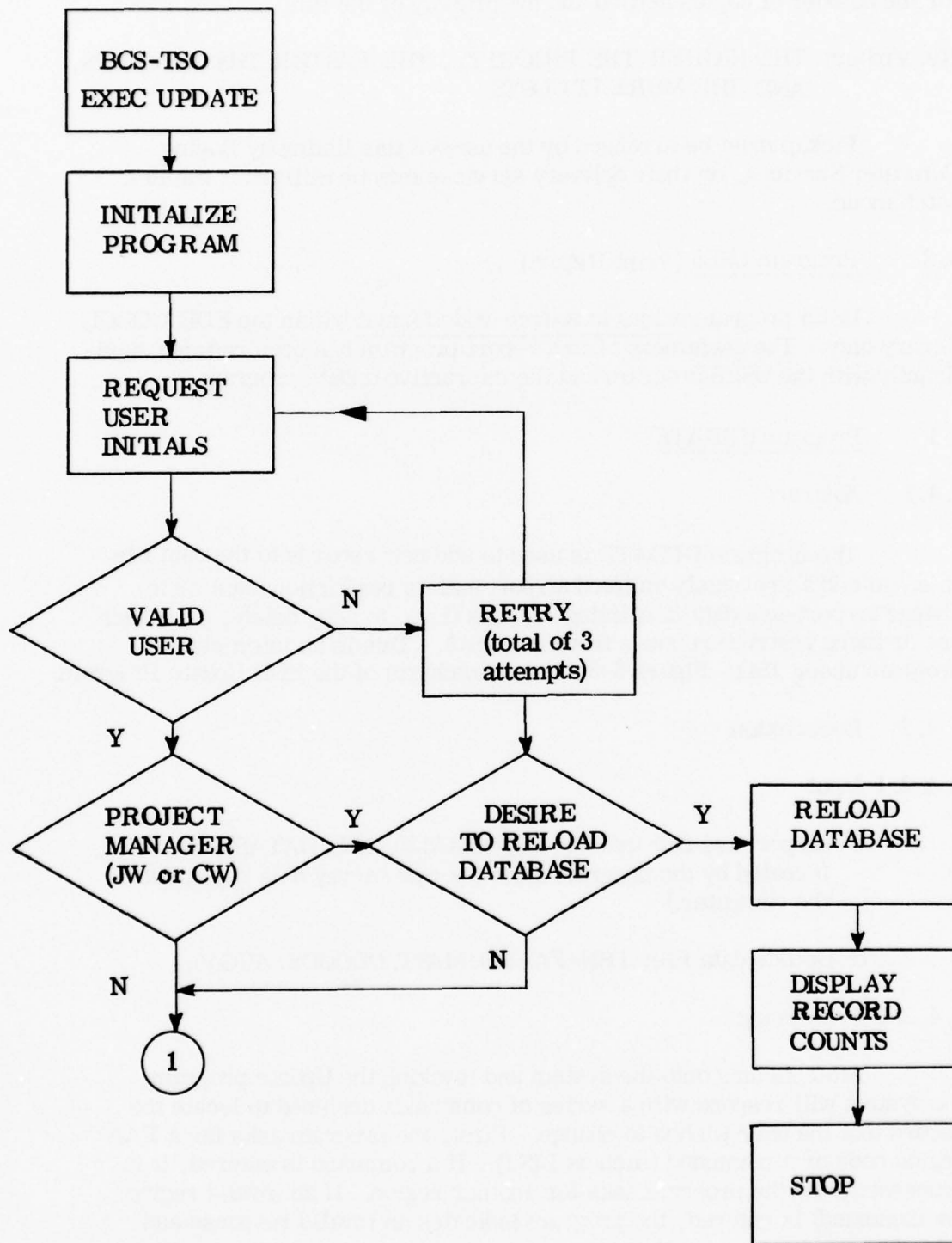


Figure 5-3. EDB Program UPDATE Flowchart
(Continued)

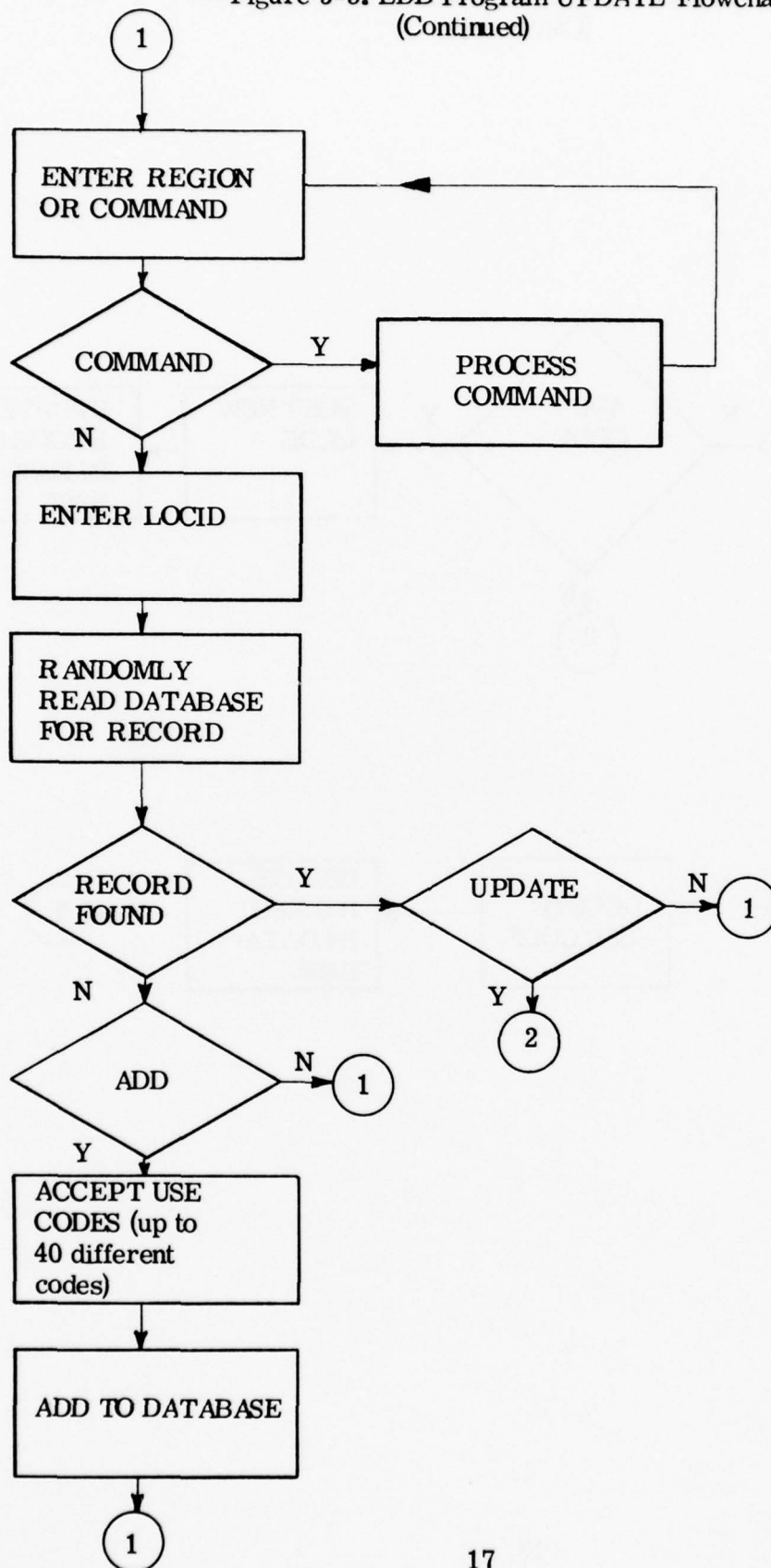
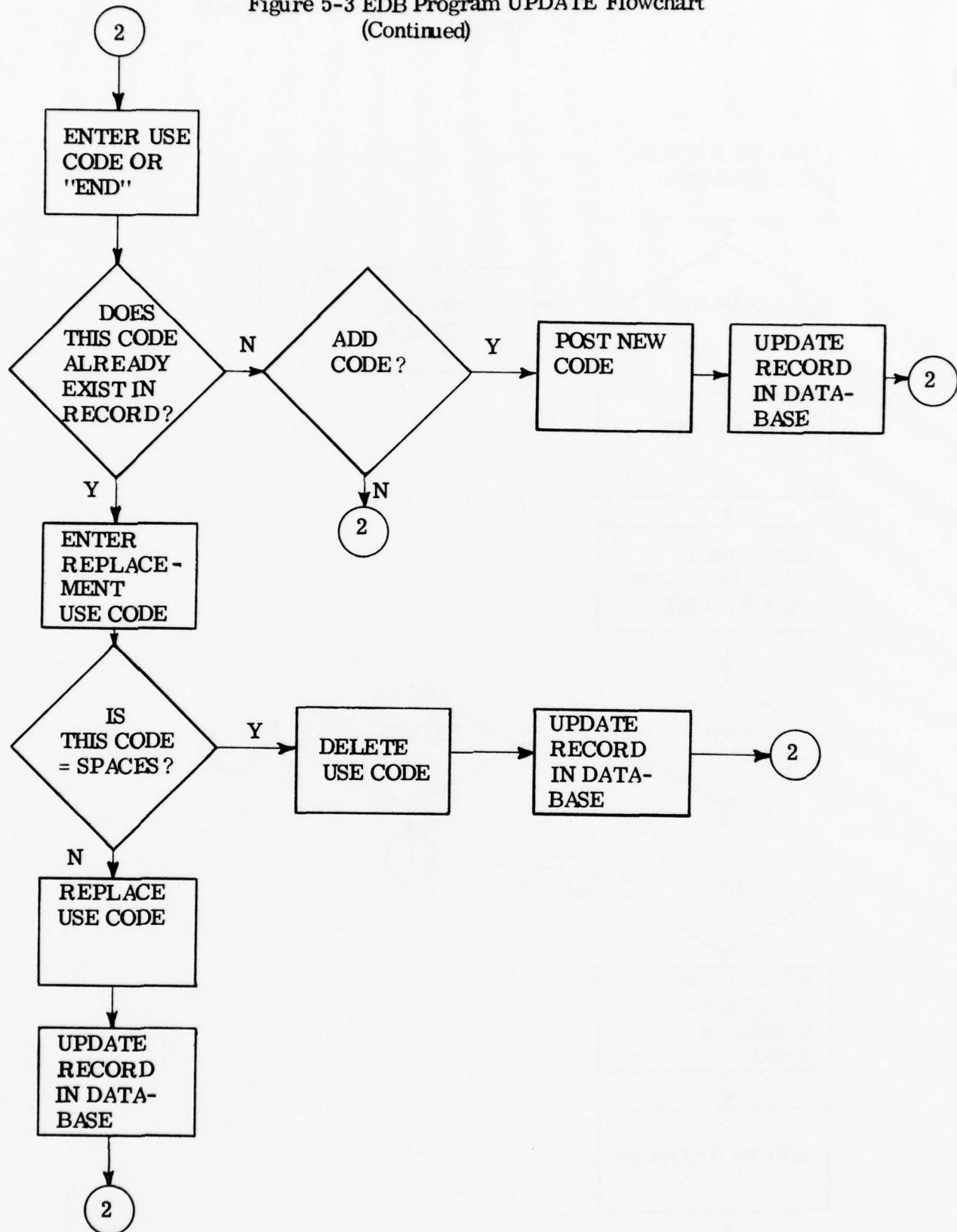


Figure 5-3 EDB Program UPDATE Flowchart
(Continued)



If a valid region is entered, the program asks for an airport location identification code (LOCID). When entered, the program randomly searches the EDB Data Bank file for that LOCID. If it is not found, the program asks the user if the record is to be added. If the user responds, NO, the program exits from the LOCID and asks for a new region. If the user responds, YES, the program allows the new record to be entered and added to the Data Bank. The program then exits from that LOCID and asks for a new region.

If, instead of entering a new LOCID, as above, the user enters one that is in the Data Bank, the program asks if the user wishes to update the existing record. The user may then add, delete, or change one or more existing use (restriction) codes by entering a new code. The program will first compare the new use code with the Decode file to insure that it is a valid code. If it is invalid, the program will so indicate and ask for another use code.

If a valid code is entered, the program compares it to the existing codes in the record. If the new code is not found, the program asks the user if the new code is to be added. If the user responds, YES, the code is added to the record and the program asks for a new use code. If the user responds, NO, the program exits and asks for a new use code.

If a valid use code is entered and the code is found in the record, the program asks the user to enter a replacement code (i. e., the program assumes that the existing code is to be replaced by a new code). The user then enters the use code to be substituted for the existing one. The program performs the substitution and asks for another use code. If the existing use code is to be deleted rather than replaced, the user enters a carriage return when the program asks for a replacement code. The program then deletes the existing use code, exits, and asks for a new use code.

The program will continue (after each addition, replacement, or deletion) to ask for new use codes until the user responds by entering the END command. The program then indicates that the update of the present record is complete and asks for a region. The process may then be repeated for another airport. To exit from the program, the user should enter STOP when the program asks for a region.

5. 4. 2. 3 Output

The output of the program is the updated Data Bank.

5.4.3 Error Handling

Errors in region, LOCID, and use code entries are caught by the program. In each case, the program compares the entry with the list of valid responses. If an invalid entry is encountered, the program responds (in the case of region or use code) by asking for a new code or (in the case of LOCID) asking if a new record is to be added to the Data Bank.

5.4.4 Program Maintenance

- o Program - UPDATE
- o Language - COBOL
- o Source Library - FAA130. EDB. COBOL(UPDATE)

5.4.5 Procedures for Executing UPDATE

A CLIST is available to execute the Update program. Figure 5-4 shows the CLIST listing. The procedure for executing the program is given below:

- o User logs onto BCS-TSO (USER-ID=FAA130).
- o User enters EXEC UPDATE.
- o Program begins.
- o User enters desired LOCID and REGION.
- o If record is not found, the user may elect to add it with one or more use codes.
- o If the record is found, the user may:
 - Change one or more use codes.
 - Delete one or more use codes.
 - Delete entire record.

This program uses the existing FAA master decode file to verify the accuracy of the use codes as they are entered.

Figure 5-4. Program UPDATE CLIST Listing

FREE ALL	(FREE EXISTING DATASETS)
ALLOC DA(*) F(SYSOUT)	(ALLOCATE TERMINAL FOR PROMPTING)
ALLOC DA(*) F(SYSIN)	(ALLOCATE TERMINAL FOR USER INPUT)
ALLOC DA('FAA110. MAST. DECODE. AUG78. DATA') F(DECODE)	(ALLOCATE DECODE FILE TO EDIT USE CODES)
ALLOC DA('FAA118. AIRPORT. EDB. MAR78. USE. EASYTREV. DATA')	(ALLOCATE USE CODE FILE) (This DSN should be changed if a new survey takes place)
CALL 'FAA130. EDB. LOAD(UPDATE)'	(START UPDATE PROGRAM)

5.5 Reports

The principal report generated by the system is a listing of the Data Bank in sorted form (by LOCID within each region). It is generated by the USE8 program. Figure 5-5 shows a sample page from this report. The report is quite lengthy; therefore, the user may prefer to route the printout to a high speed line printer rather than printing it at a terminal. The UPDATE Program lists one or more airports of any region.

Figure 5-5. Sample Computer Output of the USE8 Program.

A. Batch Mode

```

REGION: AAL          LOCID: ANL          AIRPORT NAME: ANCHORAGE INTERNATIONAL

2B1  LOCAL ZONING ORDINANCES (AIRPORT RELATED)
2D6  NOISE RELATED PHYSICAL BARRIERS AND/OR LANDSCAPING
2D7  NOISE RELATED NEW OR EXTENDED RUNWAYS
3B   AIRPORT NOISE CONTROL AND LAND USE COMPATIBILITY PLAN COMPLETED
5C1  PREFERENTIAL RUNWAY USAGE FOR NOISE ABATEMENT PURPOSES
5C3  PREFERENTIAL DEPARTURE TRACKS FOR NOISE ABATEMENT PURPOSES
6A   SCHOOLS LOCATED IN AIRPORT VICINITY
6B   HOSPITALS / NURSING HOMES OR SIMILAR FACILITIES LOCATED IN AIRPORT VICINITY
6C   RESIDENTIAL AREAS LOCATED IN AIRPORT VICINITY
6E   RELIGIOUS STRUCTURES OR SITES LOCATED IN AIRPORT VICINITY
6G   RECREATION FACILITIES LOCATED IN AIRPORT VICINITY
6H   HOTELS / MOTELS OR RESORTS LOCATED IN AIRPORT VICINITY
7E   50 TO 100 NOISE COMPLAINTS RECEIVED ANNUALLY

```

B. Terminal Output

```

REGION: AAL          LOCID: ANL          AIRPORT NAME: ANCHORAGE INTERNATIONAL

2B1  LOCAL ZONING ORDINANCES (AIRPORT RELATED)
2D6  NOISE RELATED PHYSICAL BARRIERS AND/OR LANDSCAPING
2D7  NOISE RELATED NEW OR EXTENDED RUNWAYS
3B   AIRPORT NOISE CONTROL AND LAND USE COMPATIBILITY PLAN COMPLETED
5C1  PREFERENTIAL RUNWAY USAGE FOR NOISE ABATEMENT PURPOSES
5C3  PREFERENTIAL DEPARTURE TRACKS FOR NOISE ABATEMENT PURPOSES
6A   SCHOOLS LOCATED IN AIRPORT VICINITY
6B   HOSPITALS / NURSING HOMES OR SIMILAR FACILITIES LOCATED IN AIRPORT VICINITY
6C   RESIDENTIAL AREAS LOCATED IN AIRPORT VICINITY
6E   RELIGIOUS STRUCTURES OR SITES LOCATED IN AIRPORT VICINITY
6G   RECREATION FACILITIES LOCATED IN AIRPORT VICINITY
6H   HOTELS / MOTELS OR RESORTS LOCATED IN AIRPORT VICINITY
7E   50 TO 100 NOISE COMPLAINTS RECEIVED ANNUALLY

```

```

REGION: AAL          LOCID: COU          AIRPORT NAME: CORDOVA-HALE 12 FIELD

7A   LESS THAN 5 NOISE COMPLAINTS RECEIVED ANNUALLY

```

```

REGION: AAL          LOCID: FRI          AIRPORT NAME: FAIRBANKS-INTERNATIONAL

2B1  LOCAL ZONING ORDINANCES (AIRPORT RELATED)
6A   SCHOOLS LOCATED IN AIRPORT VICINITY
6C   RESIDENTIAL AREAS LOCATED IN AIRPORT VICINITY
6G   RECREATION FACILITIES LOCATED IN AIRPORT VICINITY
7D   25 TO 50 NOISE COMPLAINTS RECEIVED ANNUALLY

```

Figure 5-6. Sample UPDATE Program Execution and Outputs

```

READY
ex update
ENTER USER INITIALS
cw

HELLO CARLTON ,WELCOME TO THE FAA/EDB INTERACTIVE DATA ENTRY/DISPLAY SYSTEM

IS AN INTRODUCTION TO THE SYSTEM NECESSARY?
no
CARLTON, DO YOU WANT VERBOSE OR TERSE COMMUNICATIONS? (V/T/?)

v

VERBOSE MODE SET

ENTER REGION (OR COMMAND)
aea

ENTER LOCID
iad
FACILITY AEA/IAD ALREADY EXISTS
WOULD YOU LIKE TO UPDATE THE RECORD? (YES/NO/?)
list
INVALID RESPONSE - LIST - REENTER
FACILITY AEA/IAD ALREADY EXISTS
WOULD YOU LIKE TO UPDATE THE RECORD? (YES/NO/?)
yes
ENTER THE USE CODE YOU WISH TO UPDATE (USE CODE), "END", "?"
end
UPDATE COMPLETED

ENTER REGION (OR COMMAND)
aea

ENTER LOCID
iad
FACILITY AEA/IAD ALREADY EXISTS
WOULD YOU LIKE TO UPDATE THE RECORD? (YES/NO/?)
ni
INVALID RESPONSE - NI - REENTER
FACILITY AEA/IAD ALREADY EXISTS
WOULD YOU LIKE TO UPDATE THE RECORD? (YES/NO/?)
no

ENTER REGION (OR COMMAND)
list
ENTER THE REGION OF THE RECORD YOU WANT LISTED
aea
ENTER LOCID
iad
REGION: AEA LOCID: IAD

2B1 LOCAL ZONING ORDINANCES (AIRPORT RELATED)
2D8 AIRPORT EQUIPPED WITH NOISE MONITORING SYSTEM
3A CONTINUING NOISE COMMITTEE AT LOCAL LEVEL
3C AIRPORT NOISE CONTROL AND LAND USE COMPATIBILITY PLAN PROPOSED OR UN
DERWAY
4B1B CURFEW FROM 2200 TO 0700
4B2C OTHER AIRCRAFT AFFECTED BY CURFEW
5A4 NOISE ABATEMENT PROCEDURE: DECELERATING
5J OTHER NOISE ABATEMENT PROCEDURES OR FLIGHT OPERATIONS RESTRICTIONS
6A SCHOOLS LOCATED IN AIRPORT VACINITY
6C RESIDENTIAL AREAS LOCATED IN AIRPORT VACINITY
6E RELIGIOUS STRUCTURES OR SITES LOCATED IN AIRPORT VACINITY
6G RECREATION FACILITIES LOCATED IN AIRPORT VACINITY
6H HOTELS / MOTELS OR RESORTS LOCATED IN AIRPORT VACINITY
7G OVER 200 NOISE COMPLAINTS RECEIVED ANNUALLY

ENTER REGION (OR COI
READY
ex update

ENTER USER INITIALS
jw

HELLO JOHN ,WELCOME TO THE FAA/EDB INTERACTIVE DATA ENTRY/DISPLAY SYSTEM

IS AN INTRODUCTION TO THE SYSTEM NECESSARY?
no
JOHN , DO YOU WANT VERBOSE OR TERSE COMMUNICATIONS? (V/T/?)

t

TERSE MODE SET

REGION?
aea
LOCID?
dca
UPDATE?
no

```

APPENDIX A

- o FAA EDB Form 1050-5
- o COBOL Coding Form

ENVIRONMENTAL DATA BANK

(RIS: EQ 1050-2)

Region

Airport

LOCID

UNDER EACH PART, PLEASE CIRCLE THE APPROPRIATE RESPONSE LETTER(a) OR NUMBER(s)

PART 1 STATE STATUTES

- A. Noise Control (airport/aircraft related)
- B. Airport Land Use Control
- C. Emissions Control (airport/aircraft related)
- D. Other Environmental Protection (airport/aircraft related)

PART 2 LOCAL GOVERNMENT ACTIONS

- A. Noise Regulations (airport/aircraft related)
- B. Land Use Control (noise related)

- 1. Zoning
 - 2. Building Codes (noise related)
- If a building code requires soundproofing of buildings in airport vicinity please indicate below:

- a) Schools
- b) Hospitals, nursing homes or similar facilities
- c) Houses or apartment buildings
- d) Office buildings
- e) Other

C. Soundproofing Programs

- 1. By schools
- 2. By hospitals, nursing homes, or similar facilities
- 3. By homes, apartment buildings
- 4. Office buildings
- 5. Other

For any of these, if government financial assistance is provided, please indicate which below:

- a) Federal
- b) State

D. Airport Actions for Environmental Protection

- 0. Land acquisition

If financial aid is provided please indicate which type below:

- a) Federal assistance
- b) State assistance

- 1. Easements on property surrounding airport
- If financial aid is provided please indicate which type below:

- a) Federal assistance
- b) State assistance

- 2. Noise tax or fee
- 3. Peak pricing
- 4. Utility expenditures/limitations
- 5. Suppressing equipment
- 6. Physical barriers, landscaping (noise related)
- 7. New or extended runways (noise related)
- 8. Noise monitoring system
- 9. Air pollution monitoring system

PART 3 NOISE CONTROL ACTIVITIES

- A. Continuing Noise Committee at Local Level
- B. Airport Noise Control and Land Use Compatibility Plan Completed
- 1. Noise control only
- 2. Land use compatibility only
- C. Airport Noise Control and Land Use Compatibility Plan Proposed or Underway
- D. Interest in Participating in a Noise Control and Land Use Compatibility Plan
- E. Have Reflected an Opportunity to Participate in a Noise Control and Land Use Compatibility Plan
- F. Other Noise Control Activities Dissimilar to Any of the Above

PART 4 LOCAL AIRPORT USE RESTRICTIONS

A. Aircraft Type or Weight Restrictions

0. All jet
1. Large air carrier type jet (75,000 lbs or over)
2. Business jet of any type
3. Multi-engine (piston, large)
4. Single-engine (piston, 1000 lbs or over)
5. CAB certificated air carrier
6. 12,500 lbs gross weight (or over)
7. 30,000 lbs gross weight (or over)
8. Helicopter
9. Other aircraft dissimilar to any of the above

B. Curfew (If applicable, please specify which period most closely matches)

1. Time
 - a. 2200 - 0600
 - b. 2200 - 0700
 - c. 2200 - 0800
 - d. 2300 - 0600
 - e. 2300 - 0700
 - f. 2300 - 0800
 - g. 2400 - 0600
 - h. 2400 - 0700
 - i. 2400 - 0800
 - j. Other period(s) dissimilar to any of the above

Please indicate below the type of aircraft affected by the curfew:

2. Aircraft Affected
 - a. Jet aircraft only
 - b. All aircraft
 - c. Other

C. Non-FAR 36 Restrictions

1. Exceptions (If applicable, please specify below)

- (a) Cessna citation
- (b) Other certain specified aircraft with "low" noise level

D. Ground Operation Restrictions

1. Location of engine runup maintenance
2. Time of engine runup maintenance
3. Preferential taxiways
4. Taxiing thrust restriction
5. Towing requirement
6. Other restrictions dissimilar to any of the above

E. Total Number of Operations Limitations for Noise Abatement Purposes

1. Year limits
2. Month limits
3. Day limits
4. Hour limits

PART 5 FLIGHT OPERATIONS - NOISE ABATEMENT PROCEDURES

- A. Reduced Thrust Approach
 - 1. Steeper angle glide slope
 - 2. Reduced flaps
 - 3. Minimum flaps
 - 4. Decelerating
 - 5. Profile descent program
 - 6. Other
- B. Glide Slope Intercept Altitude
- C. Preferential Operations
 - 1. Runways
 - 2. Approach tracks
 - 3. Departure tracks
- D. Rotational Runway System
- E. Maximum (safe) Climb on Takeoff
- F. Takeoff Thrust Reduction
- G. Reverse Thrust Reduction
- H. Displaced Thresholds (noise related)
 - 1. For takeoffs
 - 2. For landings
- I. Flight Training Restrictions
 - 1. Touch and go operations
 - 2. Time period restriction
 - 3. Day(s) of week restriction
- J. Other Procedures or Restrictions Dissimilar to Any of the Above

PART 6 SPECIAL NOISE SENSITIVE AREAS NEAR AIRPORT

- A. Schools
- B. Hospitals, Nursing Homes or Similar Facilities
- C. Residential Areas
- D. Historic Sites, National Park or Forest
- E. Religious Structures or Sites
- F. Concert Hall(s) or Public Gathering Place(s)
- G. Recreation Facilities
- H. Hotels, Motels, Resorts
- I. Other Areas Dissimilar to Any of the Above

PART 7 NUMBER OF NOISE COMPLAINTS RECEIVED ANNUALLY (approx.)

- A. Less than 5
- B. 5 - 10
- C. 10 - 25
- D. 25 - 50
- E. 50 - 100
- F. 100 - 200
- G. Over 200

PART 8 IS AIRPORT LOCATED IN AN AIR QUALITY MAINTENANCE AREA?

- A. Yes
- B. No

Form 1050-5 COBOL Coding Form

SYSTEM		PUNCHING INSTRUCTIONS		PAGE		OF	
PROGRAM		GRAPHIC PUNCH		CARD FORM #			
PROGRAMMER		DATE					

Form 1050-5 Part Number Followed By Circled Subset(s) For Each 4 Columns Block

LO- CID	# RE- # GION	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80
1	3	4	6	7																

* Card Number - 1 or 2.

APPENDIX B
PROGRAM LISTINGS

o Source Program

1. FAA130. EDB. COBOL(USE8)
2. FAA130. EDB. COBOL(SORTER)
3. FAA130. EDB. COBOL(UPDATE)
4. FAA130. EDB. COBOL(USE5)

o Job Control Language and TSO Command Lists for USE8, SORTER, LIST, and COBOL

1. FAA130. USE8. CLIST
2. FAA130. USE8. CNTL
3. FAA130. COBOL. CLIST
4. FAA130. LIST. CLIST
5. FAA130. UPDATE. CLIST
6. FAA130. SORTER. CLIST

Source Programs

1. EDB.COBOL(USE8)

```

000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID.
000030 'USE8'.
000040 AUTHOR.
000050 ROBERT STANLEY.
000060 REMARKS.
000070 ENVIRONMENT DIVISION.
000080 CONFIGURATION SECTION.
000090 SOURCE-COMPUTER. IBM-370.
000100 OBJECT-COMPUTER. IBM-370.
000110 SPECIAL-NAMES.
000120 COL IS TO-NEXT-PAGE.
000130 INPUT-OUTPUT SECTION.
000140 FILE-CONTROL.
000150 SELECT USE-IN ASSIGN TO UT-S-USEIN.
000160 SELECT USE-PRT ASSIGN TO UT-S-USEPRT.
000170 SELECT USE-OUT ASSIGN TO UT-S-USEOUT.
000180 SELECT DCOU-FILE ASSIGN TO DA-1-DCOU.
000190 RECORD KEY IS REC-KEY
000200 ACCESS IS RANDOM
000210 NOMINAL KEY IS NOM-KEY.
000220 DATA DIVISION.
000230 FILE SECTION.
000240 FD USE-IN
000250 RECORDING MODE IS F
000260 BLOCK CONTAINS 0 RECORDS
000270 RECORD CONTAINS 90 CHARACTERS
000280 LABEL RECORDS ARE STANDARD
000290 DATA RECORD IS USEIN.
000300 01 USEIN PIC X(90).
000310 FD USE-PRT
000320 BLOCK CONTAINS 1 RECORDS
000330 RECORD CONTAINS 133 CHARACTERS
000340 RECORDING MODE IS F
000350 LABEL RECORDS ARE STANDARD
000360 DATA RECORD IS PRT-REC.
000370 01 PRT-REC PIC X(133).
000380 FD DCOU-FILE
000390 RECORDING MODE IS F
000400 BLOCK CONTAINS 10 RECORDS
000410 LABEL RECORDS ARE STANDARD
000420 RECORD CONTAINS 160 CHARACTERS
000430 DATA RECORD ARE DCOU-REC DCOU2-REC.
000440 01 DCOU-REC.
000450 05 FILLER PIC X.
000460 05 REC-KEY PIC X(19).
000470 05 APMAME PIC X(42).
000480 05 FILLER PIC X(98).
000490 01 DCOU2-REC.
000500 05 FILLER PIC X(20).
000510 05 USERCQUE PIC X(96).
000520 05 FILLER PIC X(44).
000530 FD USE-OUT
000540 RECORDING MODE IS F
000550 LABEL RECORDS ARE STANDARD
000560 BLOCK CONTAINS 1 RECORDS
000570 RECORD CONTAINS 153 CHARACTERS
000580 DATA RECORD IS USEOUT.
000590 01 USEOUT.
000600 02 FIL PIC X(153).
000610 WORKING-STORAGE SECTION.

```

000620	77	REG-HLD	PIC XXX VALUE SPACES.
000630	77	SUB	PIC 99 VALUE ZEROS.
000640	77	ERR-SW	PIC 9 VALUE ZEROS.
000650	77	SUB2	PIC 99 VALUE ZEROS.
000660	77	LOC-HOLD	PIC XXX VALUE SPACES.
000670	77	NON-KEY	PIC X(19).
000680	01	FIRST-SW PIC X VALUE '0'.	
000690	01	BLK-LINE SYNC.	
000700	05	FILLER	PIC X(133) VALUE SPACES.
000710	01	ASTR-LINE SYNC.	
000720	05	FILLER	PIC X(120) VALUE ALL '0'.
000730	01	KEY-HLD SYNC.	
000740	02	KEY1.	
000750	05	FILLER	PIC X(8) VALUE 'ARPTNAME'.
000760	05	LOC	PIC XXX.
000770	05	FILLER	PIC X(8) VALUE SPACES.
000780	02	KEY2.	
000790	05	FILLER	PIC X(8) VALUE 'USERESCD'.
000800	05	UCDE	PIC XXXX.
000810	05	FILLER	PIC X(7) VALUE SPACES.
000820	01	USE-HLD SYNC.	
000830	05	LOCID	PIC X(3).
000840	05	USENUM	PIC X.
000850	05	REG	PIC X(3).
000860	05	USE-CODE OCCURS 18 TIMES	PIC XXXX.
000870	05	FILLER	PIC X(11).
000880	01	WS-USE SYNC.	
000890	05	FILLER	PIC X.
000900	05	REGPT	PIC XXX.
000910	05	FILLER	PIC X.
000920	05	LOCPT	PIC XXX.
000930	05	FILLER	PIC X.
000940	05	CDEPT	PIC XXXX.
000950	05	FILLER	PIC X.
000960	05	TEXTPT	PIC X(196).
000970	05	FILLER	PIC X.
000980	05	APNAMEPT	PIC X(42).
000990	01	PRT1 SYNC.	
001000	05	FILLER	PIC X(10) VALUE SPACES.
001010	05	FILLER	PIC X(8) VALUE 'REGION'.
001020	05	PRTREG	PIC XXX.
001030	05	FILLER	PIC X(9) VALUE SPACES.
001040	05	FILLER	PIC X(7) VALUE 'LOCID'.
001050	05	PRT-LOC	PIC XXX.
001060	05	FILLER	PIC X(13) VALUE SPACES.
001070	05	FILLER	PIC X(14) VALUE 'AIRPORT NAME'.
001080	05	PRT1-APNAME	PIC X(42).
001090	05	FILLER	PIC X(24) VALUE SPACES.
001100	01	PRT2 SYNC.	
001110	05	FILLER	PIC X(3).
001120	05	PRTUSE	PIC XXXX.
001130	05	FILLER	PIC XXX.
001140	05	PRT-USE	PIC X(196).
001150	05	FILLER	PIC X(27).
001160	01	WS-ACCUMMS SYNC.	
001170	05	IM-CNT	PIC 999 VALUE ZEROS.
001180	05	PRT-CNT	PIC 9999 VALUE ZEROS.
001190	05	UCDU-CNT	PIC 9(5) VALUE ZEROS.
001200	05	OUT-CNT	PIC 999 VALUE ZEROS.
001210	01	LMAGE SECTION.	
001220	01	PARM-DATA.	

```

001230 02 FILLER PIC XX.
001240 02 PARM-INPUT PIC X(5).
001250 PROCEDURE DIVISION USING PARM-DATA.
001260 OPEN-FILES.
001270 OPEN INPUT USE-IN DCOO-FILE
001280 OUTPUT USE-PRT USE-CUT.
001290 MOVE SPACES TO WS-USE USE-HLD.
001300 READ-TRANS.
001310 READ USE-IN INTO USE-HLD AT END GO TO EOL.
001320 ADD 1 TO IN-CNT.
001330 MOVE SPACES TO PRT2.
001340 IF LOCID = LOC-HOLD MOVE ZEROS TO SUB
001350 GO TO LOAD-LINES.
001360 MOVE LOCID TO LOC PRT-LOC LOC-HOLD LOCPT.
001370 WRITE PRT-REC FROM BLK-LINE AFTER ADVANCING TO-NEXT-PAGE.
001380 MOVE REG TO REGPT.
001390 MOVE KEY1 TO REC-KEY NON-KEY.
001400 READ DCOO-FILE INVALID KEY GO TO LCUU-ERR.
001410 ADD 1 TO DCOO-CNT.
001420 CONT.
001430 IF ERA-SM = 1 MOVE 'AIRPORT NAME NOT ON DECODE FILE'
001440 TO PRT1-APNAME MOVE 0 TO ERR-SM
001450 ELSE
001460 MOVE APNAME TO PRT1-APNAME APNAMEPT.
001470 MOVE REG TO PRTREG.
001480 IF PARM-INPUT = 'BATCH' MOVE SPACES TO
001490 ASTR-LINE ELSE MOVE ALL ' ' TO ASTR-LINE.
001500 IF FIRST-SM = '0' MOVE SPACES TO ASTR-LINE
001510 MOVE '1' TO FIRST-SM.
001520 IF REG = REG-HLD
001530 WRITE PRT-REC FROM BLK-LINE AFTER ADVANCING 1 LINES
001540 WRITE PRT-REC FROM ASTR-LINE AFTER ADVANCING 2 LINES
001550 WRITE PRT-REC FROM BLK-LINE AFTER ADVANCING 1 LINES
001560 WRITE PRT-REC FROM PRT1 AFTER ADVANCING 2 LINES
001570 WRITE PRT-REC FROM BLK-LINE AFTER ADVANCING 1 LINES
001580 ADD 3 TO PRT-CNT
001590 ELSE
001600 WRITE PRT-REC FROM BLK-LINE AFTER ADVANCING 1 LINES
001610 WRITE PRT-REC FROM ASTR-LINE AFTER ADVANCING 1 LINES
001620 WRITE PRT-REC FROM BLK-LINE AFTER ADVANCING 1 LINES
001630 WRITE PRT-REC FROM PRT1 AFTER ADVANCING 2 LINES
001640 MOVE REG TO REG-HLD
001650 WRITE PRT-REC FROM BLK-LINE AFTER ADVANCING 1 LINES
001660 ADD 2 TO PRT-CNT.
001670 MOVE ZEROS TO SUB.
001680 LOAD-LINES.
001690 ADD 1 TO SUB.
001700 IF SUB > 18 GO TO READ-TRANS.
001710 IF (USE-CODE (SUB) = SPACES) OR (SUB = 19)
001720 GO TO READ-TRANS.
001730 MOVE USE-CODE (SUB) TO UCDE.
001740 MOVE KEY2 TO REC-KEY NON-KEY.
001750 READ DCOO-FILE INVALID KEY GO TO DCOU-ERR2.
001760 ADD 1 TO DCOO-CNT.
001770 CONT2.
001780 MOVE SPACES TO PRT2.
001790 MOVE USE-CODE (SUB) TO PRTUSE CODEPT.
001800 MOVE USERCODE TO PRT-USE TEXTPT.
001810 WRITE PRT-REC FROM PRT2 AFTER ADVANCING 1 LINES.
001820 ADD 1 TO PRT-CNT.
001830 WRITE USEOUT FROM WS-USE.

```

```

001830      ADD 1 TO OUT-CNT.
001840      GO TO LOAD-LINES.
001850  EQU.
001860      DISPLAY 'RECS READ IN: ' IN-CNT.
001870      DISPLAY 'RECS PRINTED: ' PRT-CNT.
001880      DISPLAY 'RECS DCODED: ' DCOD-CNT.
001890      DISPLAY 'RECS CREATED: ' OUT-CNT.
001900      CLOSE USE-IN DCOD-FILE USE-PRT USE-OUT.
001910      STOP RUN.
001920  DCOD-ERR.
001930      DISPLAY 'INVALID KEY: ' KEY1.
001940      MOVE 1 TO ERR-SM.
001950      GO TO CONT.
001960  DCOD-ERR2.
001970      DISPLAY '*** ERROR - BAD USE CODE - ' USE-CODE (SUB
001980      GO TO CONT2.

```



```

2. EDB, COBOL (SORTER)
00010 IDENTIFICATION DIVISION.
00020 PROGRAM-ID. 'SORTER'.
00030 ENVIRONMENT DIVISION.
00040 CONFIGURATION SECTION.
00050 INPUT-OUTPUT SECTION.
00060 FILE-CONTROL.
00070 SELECT SORT-FILE ASSIGN TO UT-S-DUMMY.
00080 SELECT RAW-DATA ASSIGN TO UT-S-RAWINPUT.
00090 SELECT DATABASE ASSIGN TO DA-M-DATABASE.
00100 ACCESS IS SEQUENTIAL ACTUAL KEY IS ACTUAL-KEY.
00110 DATA DIVISION.
00120 FILE SECTION.
00130 SD SORT-FILE.
00140 01 SORT-REC.
00150 02 SORT-LOCID PIC X(3).
00160 02 SORT-CARDTYPE PIC X.
00170 02 SORT-REGION PIC X(3).
00180 02 SORT-USES.
00190 03 SORT-USE-CODE OCCURS 18 TIMES PIC X(4).
00200 02 SORT-FILLER PIC X(11).
00210 PD RAW-DATA LABEL RECORDS ARE STANDARD RECORD 90 BLOCK 0.
00220 01 RAW-RECORD.
00230 02 RAW-LOCID PIC X(3).
00240 02 RAW-CARDTYPE PIC X.
00250 02 RAW-REGION PIC X(3).
00260 02 RAW-USE-CODES OCCURS 18 TIMES PIC X(4).
00270 02 FILLER PIC X(3).
00280 02 RAW-TSO-LINE-NUM PIC 9(8).
00290 PD DATABASE LABEL RECORDS ARE STANDARD RECORD 167.
00300 01 DATA-REC.
00310 02 DATA-REC-KEY.
00320 03 DATA-REC-REGION PIC X(3).
00330 03 DATA-REC-LOCID PIC X(3).
00340 02 DELETE-FLAG PIC X.
00350 02 USES.
00360 03 USE-CODE-ARRAY OCCURS 40 TIMES PIC X(4).
00370 02 FILLER REDEFINES USES.
00380 04 FILLER PIC X(72).
00390 04 TYPE2-REC PIC X(72).
00400 04 FILLER PIC X(16).
00410 EJECT
WORKING-STORAGE SECTION.
00420 01 COUNTERS.
00430 02 INS PIC S9(5) COMP-3 VALUE ZERO.
00440 02 TYPE1 PIC S9(5) COMP-3 VALUE ZERO.
00450 02 TYPE2 PIC S9(5) COMP-3 VALUE ZERO.
00460 02 TSO-LINE-NUM PIC S9(8) COMP-3 VALUE ZERO.
00470 01 ACTUAL-KEY.
00480 02 TRACK-ID PIC S9(5) COMP VALUE +0.
00490 02 RECORD-KEY.
00500 03 KEY-REGION PIC X(3).
00510 03 KEY-LOCID PIC X(3).
00520 02 RECORD-KEY2 REDEFINES RECORD-KEY.
00530 03 FILLER PIC X.
00540 88 DUMMY-RECORD VALUE HIGH-VALUES.
00550 03 FILLER PIC X(5).
00560 01 KEY-HILO SYNC.
00570 02 KEY1.
00580

```

```

00590      03 FILLER PIC X(8) VALUE 'ARPTNAME'.
00600      03 FILLER PIC X(8) VALUE SPACES.
00610
00620      02 KEY2.
00630      03 FILLER PIC X(8) VALUE 'USERSCD'.
00640      03 UCDEF PIC X(4).
00650      03 FILLER PIC X(7) VALUE SPACES.
00660
00670      01 NON-KEY PIC X(19) VALUE SPACES.
00680      PROCEDURE DIVISION.
00690          MOVE 'SORTMSG' TO SORT-MESSAGE.
00700          MOVE ALL '9' TO SORT-CORE-SIZE.
00710      SORT-USERS.
00720          MOVE ZERO TO SORT-RETURN.
00730          SORT SORT-FILL ON ASCENDING KEY
00740              SORT-REGION SORT-LOCID SORT-CARDTYPE
00750              INPUT PROCEDURE IS BEFORE-THE-SORT
00760              OUTPUT PROCEDURE IS AFTER-THE-SORT.
00770          IF SORT-RETURN NOT = ZERO
00780              DISPLAY 'UNSUCCESSFUL SORT FOR USERS DATA'
00790              MOVE 16 TO RETURN-CODE.
00800          DISPLAY 'INPUT = ' INS.
00810          DISPLAY 'TYPE1 = ' TYPE1.
00820          DISPLAY 'TYPE2 = ' TYPE2.
00830          STOP RUN.
00840      BEFORE-THE-SORT SECTION.
00850          LOAD-SEQ.
00860          OPEN INPUT DATABASE.
00870          LOAD-01.
00880          READ DATABASE AT END CLOSE DATABASE GO TO
00890              BEFORE-THE-SORT-EXIT.
00900          IF DUMMY-RECORD OR DELETE-FLAG = '1' GO TO LOAD-01.
00910          ADD 1 TO INS.
00920          PERFORM REFORMAT-DATA THRU REFORMAT-DATA-EXIT.
00930          GO TO LOAD-01.
00940      LOAD-SEQ-EXIT. EXIT.
00950      REFORMAT-DATA.
00960          GNE-TYPE1.
00970          MOVE DATA-REC-REGION TO SORT-REGION.
00980          MOVE DATA-REC-LOCID TO SORT-LOCID.
00990          MOVE '1' TO SORT-CARDTYPE.
01000          MOVE USES TO SORT-USERS.
01010          RELEASE SORT-REC.
01020          ADD 1 TO TYPE1.
01030          IF USE-CC'E-ARRAY (19) = SPACES GO TO REFORMAT-DATA-EXIT.
01040          GEN-TYPE2.
01050          MOVE DATA-REC-REGION TO SORT-REGION.
01060          MOVE DATA-REC-LOCID TO SORT-LOCID.
01070          MOVE '2' TO SORT-CARDTYPE.
01080          MOVE TYPE2-REC TO SORT-USERS.
01090          RELEASE SORT-REC.
01100          ADD 1 TO TYPE2.
01110          REFORMAT-DATA-EXIT. EXIT.
01120          BEFORE-THE-SORT-EXIT. EXIT.
01130          AFTER-THE-SORT SECTION.
01140          OPEN OUTPUT RAW-DATA.
01150          RETURN SORT-FILE AT END CLOSE RAW-DATA GO TO
01160              AFTER-THE-SORT-EXIT.
01170          MOVE SORT-REC TO RAW-RECORD.
01180          ADD 10 TO TSO-LINE-NUM.
01190          MOVE TSO-LINE-NUM TO RAW-TSO-LINE-NUM.
01200          WRITE RAW-RECORD.
01210          GO TO AFTER-01.
01220      AFTER-THE-SORT-EXIT.
01230          EXIT.
01240      READY

```

3. EDB.COBOL(UPDATE)

```

000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID. UPDATE.
000030 AUTHOR. TRANS SYSTEMS CORP.
000040 ENVIRONMENT DIVISION.
000050 CONFIGURATION SECTION.
000060 SOURCE=COMPUTER. IBM-370.
000070 OBJECT=COMPUTER. IBM-370.
000080 SPECIAL-NAMES.
000090 INPUT-OUTPUT SECTION.
000100 FILE-CONTROL.
000110 SELECT UCOD-FILE ASSIGN TO DA-I-UCOD
000120 RECORD-KEY IS REC-KEY
000130 ACCESS IS RANDOM
000140 NOMINAL KEY IS NON-KEY.
000150 SELECT SORT-FILE ASSIGN TO UT-S-DUMMY.
000160 SELECT RAW-DATA ASSIGN TO UT-S-RAWINPUT.
000170 SELECT DATABASE ASSIGN TO DA-W-DATABASE
000180 ACCESS IS RANDOM ACTUAL KEY IS ACTUAL-KEY.
000190 DATA DIVISION.
000200 FILE SECTION.
000210 SQ SORT-FILE.
000220 01 SORT-REC.
000230 02 SORT-USE-CODE PIC X(4).
000240 02 FILLER PIC X(11).
000250 FD UCOD-FILE
000260 RECORDING MODE IS F
000270 BLOCK CONTAINS 10 RECORDS
000280 LABEL RECORDS ARE STANDARD
000290 RECORD CONTAINS 160 CHARACTERS.
000300 01 UCOD-REC.
000310 02 FILLER PIC X.
000320 02 REC-KEY PIC X(11).
000330 02 APNAME PIC X(42).
000340 02 FILLER PIC X(98).
000350 01 UCOD2-REC.
000360 02 FILLER PIC X(20).
000370 02 USERCODE PIC X(96).
000380 02 FILLER PIC X(4).
000390 FD RAW-DATA
000400 LABEL RECORDS ARE STANDARD
000410 RECORD 90 BLOCK 0.
000420 01 RAW-RECORD.
000430 02 RAW-LOCID PIC X(13).
000440 02 RAW-CARDTYPE PIC X.
000450 02 RAW-REGION PIC X(13).
000460 02 RAW-USE-CODES OCCURS 18 TIMES PIC X(4).
000470 02 FILLER PIC X(11).
000480 FD DATABASE
000490 LABEL RECORDS ARE STANDARD
000500 RECORD CONTAINS 167 CHARACTERS.
000510 01 DATA-REC.
000520 02 DATA-REC-KEY.
000530 03 DATA-REC-REGION PIC X(13).
000540 03 DATA-REC-LOCID PIC X(13).
000550 02 DELETE-FLAG PIC X.
000560 02 USES.
000570 03 USE-CODE-ARRAY OCCURS 40 TIMES PIC X(4).
000580 EJECT
000590 WORKING-STORAGE SECTION.
000600 01 KEY-HLD SYNC.
000610 02 KEY1.

```

000620 03 FILLER PIC X(8) VALUE 'ARFTNAME'.
000630 03 LOC PIC XXX.
000640 03 FILLER PIC X(8) VALUE SPACES.
000650 02 KEYZ.
000660 03 FILLER PIC X(8) VALUE 'USERESCO'.
000670 03 CODE PIC X(4).
000680 03 FILLER PIC X(7) VALUE SPACES.
000690 01 NON-KEY PIC X(19) VALUE SPACES.
000700 01 SORT-SUB PIC S9(4) COMP VALUE +0.
000710 01 DUP-USE-CODE PIC X(4) VALUE SPACES.
000720 01 HOLD-CODE-TABLE.
000730 02 HOLD-CODE-ARRAY OCCURS 40 TIMES PIC X(4).
000740 01 REGION-OK PIC X VALUE '0'.
000750 88 INVALID-REGION VALUE '1'.
000760 01 LOCID-OK PIC X VALUE '0'.
000770 88 INVALID-LOCID VALUE '1'.
000780 01 CODE-OK PIC X VALUE '0'.
000790 88 INVALID-USE-CODE VALUE '1'.
000800 01 INVALID-KEY-SM PIC X VALUE '0'.
000810 88 INVALID-KEY VALUE '1'.
000820 01 END-SM PIC X VALUE '0'.
000830 01 IN-USE-CODE.
000840 02 EDIT-USE-CODE PIC X(4) VALUE SPACES.
000850 02 FILLER PIC X VALUE SPACES.
000860 01 OLD-USE-CODE PIC X(4) VALUE SPACES.
000870 01 RECORD-HOLD-AREA.
000880 02 ARRAY-RECORD OCCURS 2 TIMES.
000890 03 ARRAY-LOCID PIC X(3).
000900 03 ARRAY-CARDTYPE PIC X.
000910 03 ARRAY-REGION PIC X(3).
000920 03 ARRAY-USE-CODE OCCURS 18 TIMES PIC X(4).
000930 01 USE-SUB PIC S9(4) COMP VALUE ZERO.
000940 01 FINISHED-SM PIC X VALUE '0'.
000950 01 END-FLAG PIC X VALUE '0'.
000960 01 EXIT-FLAG PIC X VALUE '0'.
000970 01 MORE-UPDATES PIC X VALUE '0'.
000980 88 NO-MORE-UPDATES VALUE '1'.
000990 01 EOF-SM PIC X VALUE '0'.
001000 01 RAW-INPUT-COUNT PIC S9(5) COMP-3 VALUE ZERO.
001010 01 SHORT-BUFFER PIC X(10) VALUE SPACES.
001020 01 FILLER REDEFINES SHORT-BUFFER.
001030 02 CHAR OCCURS 10 TIMES PIC X.
001040 01 OUTPUT-RECORDS PIC S9(5) COMP-3 VALUE ZERO.
001050 01 REC-SUB PIC S9(4) VALUE ZERO.
001060 01 OUT-SUB PIC S9(4) VALUE ZERO.
001070 01 IN-SUB PIC S9(4) VALUE ZERO.
001080 01 CODE-COUNT PIC S9(5) COMP-3 VALUE ZERO.
001090 01 LOCAT-SUB PIC S9(4) COMP VALUE ZERO.
001100 01 LIST-SUB PIC S9(4) COMP VALUE ZERO.
001110 01 FIELD-LENGTH PIC S9(5) COMP-3 VALUE ZERO.
001120 01 IN-KEY.
001130 02 IN-REG PIC X(4).
001140 02 IN-REG2 REDEFINES IN-REG.
001150 03 EDIT-REGION PIC X(3).
001160 03 DISPLAY-SLASH PIC X(1).
001170 02 IN-LOCID.
001180 03 EDIT-LOCID PIC X(3).
001190 03 FILLER PIC X(1).
001200 01 COMMAND-FOUND PIC X VALUE '0'.
001210 88 COMMAND VALUE '1'.
001220 01 YES-NO PIC X(4).


```

001230 88 YLS VALUE 'Y' YES'.
001240 88 NOPE VALUE 'NO' 'N'.
001250 88 QUESTION VALUE '7'.
001260 01 OP-TYPE PIC XXX VALUE SPACES.
001270 01 USER-CONTROL-BLOCK.
001280 02 USER-INITIALS PIC XXX VALUE SPACES.
001290 88 RMT VALUE 'RMT'.
001300 02 USER-ID PIC X(17) VALUE SPACES.
001310 02 USER-SYSTEMS PIC X VALUE SPACES.
001320 88 SYSTEMS VALUE '1'.
001330 02 USER-ADD-UPDATE PIC X.
001340 88 DX-TU-ADD-UPDATE VALUE '1'.
001350 02 USER-SUB PIC S99 COMP VALUE +0.
001360 01 COMM-MODE PIC X VALUE 'V'.
001370 88 VERBOSE VALUE 'V'.
001380 88 TERSE VALUE 'T'.
001390 01 MESSAGES.
001400 02 M1 PIC X(19) VALUE 'INVALID RESPONSE - '.
001410 02 M2 PIC X(10) VALUE ' - REENTER'.
001420 01 ACTUAL-KEY.
001430 02 TRACK-ID PIC S9(5) COMP VALUE +0.
001440 02 RECORD-KEY.
001450 03 KEY-REGION PIC XXX.
001460 03 KEY-LOCID PIC XXX.
001470 02 BINARY-KEY REDEFINES RECORD-KEY.
001480 03 BIN1 PIC S9(5) COMP.
001490 03 BIN2 PIC S9(2) COMP.
001500 01 LEGAL-USERS.
001510 02 FILLER PIC X(12) VALUE 'RMTB08 11'.
001520 02 FILLER PIC X(12) VALUE 'JH JOHN'.
001530 02 FILLER PIC X(12) VALUE 'CM CARLTON'.
001540 02 FILLER PIC X(12) VALUE '////////////////'.
001550 02 FILLER PIC X(12) VALUE '////////////////'.
001560 02 FILLER PIC X(12) VALUE '////////////////'.
001570 02 FILLER PIC X(12) VALUE '////////////////'.
001580 02 FILLER PIC X(12) VALUE '////////////////'.
001590 02 FILLER PIC X(12) VALUE '////////////////'.
001600 02 FILLER PIC X(12) VALUE '////////////////'.
001610 02 FILLER PIC X(12) VALUE '////////////////'.
001620 01 USER-ID-TABLE REDEFINES LEGAL-USERS.
001630 02 USERS OCCURS 12 TIMES.
001640 03 1-USER-INITIALS PIC XXX.
001650 03 1-USER-NAME PIC X(17).
001660 03 1-SYSTEMS-PRIV PIC X.
001670 03 1-ADD-UPDATE PIC X.
001680 01 LOGON-WORK-AREA.
001690 02 LOGON-ERROR-SW PIC X VALUE '0'.
001700 02 LOGON-COUNT PIC 9 VALUE 0.
001710 01 RECORD-COUNTERS.
001720 02 ADD-COUNT PIC S9(4) COMP-3 VALUE ZERO.
001730 02 UPDATE-COUNT PIC S9(4) COMP-3 VALUE ZERO.
001740 02 OUTPUT-COUNT PIC S9(5) COMP-3 VALUE ZERO.
001750 02 COUNT-MASK PIC Z2,229.
001760 EJECT.
001770 PROCEDURE DIVISION.
001780 MOVE 'SORTMSG' TO SORT-MESSAGE.
001790 MAINLINE-ROUTINE.
001800 PERFORM LOGON-RTN THRU LOGON-RTN-EXIT.
001810 PERFORM INTRO-SYSTEM THRU INTRO-SYSTEM-EXIT.
001820 PERFORM SET-VERBOSITY THRU SET-VERBOSITY-EXIT.
001830 IF SYSTEMS PERFORM SPECIAL-OPEN THRU SPECIAL-OPEN-EXIT.

```



```

001840 PERFORM OPEN-RTN THRU OPEN-RTN-EXIT.
001850 PERFORM PROCESS-DATA THRU PROCESS-DATA-EXIT
001860 UNTIL NO-MORE-UPDATES.
001870 PERFORM TERMINATE-RTN THRU TERMINATE-RTN-EXIT.
001880 STOP RUN.
001890 LOGON-RTN.
001900 DISPLAY 'ENTER USER INITIALS'.
001910 ACCEPT USER-INITIALS.
001920 PERFORM LOGON-LOOKUP THRU LOGON-LOOKUP-EXIT.
001930 IF LOGON-ERROR-SW = 1.
001940 DISPLAY 'USER INITIALS ' USER-INITIALS
001950 ' ARE INVALID - REENTER!
001960 GO TO LOGON-RTN.
001970
001980 DISPLAY 'HELLO ' USER-ID
001990 ' WELCOME TO THE FAA/EDB INTERACTIVE DATA ENTRY'
002000 '/DISPLAY SYSTEM'.
002010 DISPLAY ' '.
002020 DISPLAY ' '.
002030 LOGON-RTN-EXIT. EXIT.
002040 LOGON-LOOKUP.
002050 ADD 1 TO LOGON-COUNT.
002060 IF LOGON-COUNT > 3 DISPLAY 'TOO MANY ATTEMPTS -'
002070 ' GOODBYE!
002080 STOP RUN.
002090 MOVE 1 TO USER-SUB.
002100 LOGON-OL.
002110 IF USER-INITIALS = T-USER-INITIALS (USER-SUB)
002120 MOVE ZERO TO LOGON-ERROR-SW
002130 MOVE USERS (USER-SUB) TO USER-CONTROL-BLOCK
002140 GO TO LOGON-LOOKUP-EXIT.
002150 ADD 1 TO USER-SUB.
002160 IF USER-SUB > 12
002170 MOVE '1' TO LOGON-ERROR-SW
002180 GO TO LOGON-LOOKUP-EXIT ELSE
002190 GO TO LOGON-OL.
002200 LOGON-LOOKUP-EXIT. EXIT.
002210 EJECT
002220 INTRO-SYSTEM.
002230 DISPLAY 'IS AN INTRODUCTION TO THE SYSTEM NECESSARY?'.
002240 ACCEPT YES-NO.
002250 IF YES NEXT SENTENCE ELSE GO TO INTRO-SYSTEM-EXIT.
002260 DISPLAY ' '.
002270 DISPLAY 'THE FAA/EDB DATA ENTRY/DISPLAY SYSTEM (FEODS) IS'
002280 ' DESIGNED TO ALLOW THE USER'
002290 ' TO ADD NEW RECORDS OR MODIFY EXISTING '
002300 ' RECORDS IN THE FEODS DATABASE.'.
002310 DISPLAY ' '.
002320 DISPLAY 'THIS IS ACCOMPLISHED BY ENTERING THE '
002330 ' DISPLAY 'REGION AND LOGID OF THE DESIRED FACILITY'.
002340 DISPLAY ' '.
002350 DISPLAY 'IF THE DATA ALREADY EXISTS YOU MAY ELECT TO'
002360 ' ADD OR UPDATE'.
002370 DISPLAY 'ONE OR MORE USE CODES IN THE DATA.'.
002380 DISPLAY ' '.
002390 DISPLAY 'IF IT IS NOT FOUND YOU MAY ELECT TO ADD THE '
002400 ' FACILITY TO THE SYSTEM.'.
002410 DISPLAY ' '.
002420 DISPLAY ' '.
002430 DISPLAY ' '.
002440 DISPLAY ' '.

```

*** SPECIAL FEATURES ***.

```

002450 DISPLAY ' '
002460 DISPLAY 'WHEN FIRST ADDING A FACILITY TO THE SYSTEM '
002470 DISPLAY 'YOU MAY ENTER THE KEYWORD "END" AT ANY TIME '
002480 DISPLAY 'AFTER YOU HAVE ENTERED AT LEAST ONE VALID USE '
002490 'CODE'.
002500 DISPLAY ' '
002510 DISPLAY 'THIS RESPONSE INDICATES THAT YOU HAVE SUPPLIED ALL '
002520 DISPLAY 'OF THE INFORMATION THAT YOU HAVE AVAILABLE AT THE '
002530 'PRESENT TIME.'
002540 DISPLAY ' '
002550 DISPLAY 'THE ITEM WILL BE ADDED TO THE DATABASE AT '
002560 'THIS TIME.'
002570 DISPLAY ' '
002580 DISPLAY ' '
002590 DISPLAY 'THE KEYWORD "EXIT" MEANS THAT YOU WISH TO '
002600 'SCRATCH'
002610 DISPLAY 'THE CURRENT OPERATION (EITHER ADD OR UPDATE) '
002620 DISPLAY 'IF USED DURING AN ADD OPERATION THE '
002630 '"EXIT" COMMAND'
002640 DISPLAY 'WILL CAUSE THE ENTIRE FACILITY TO BE SCRATCHED.'
002650 DISPLAY ' '
002660 DISPLAY ' '
002670 DISPLAY 'IF USED IN AN UPDATE THE REQUESTED OPERATION '
002680 DISPLAY 'WILL NOT BE PERFORMED - HOWEVER THE ORIGINAL DATA '
002690 DISPLAY 'WILL STILL BE RETAINED IN THE DATABASE.'
002700 DISPLAY ' '
002710 DISPLAY 'THE LETTER "Y" MAY BE USED INSTEAD OF "YES" '
002720 DISPLAY 'LIKEWISE, THE LETTER "N" MAY BE USED FOR "NO" '
002730 DISPLAY ' '
002740 DISPLAY ' '
002750 DISPLAY ' *** COMMANDS *** '
002760 DISPLAY ' '
002770 DISPLAY 'WHENEVER YOU ARE PROMPTED FOR A "REGION" '
002780 DISPLAY 'YOU MAY ALSO ENTER ONE OF A SERIES OF COMMANDS.'
002790 DISPLAY ' '
002800 DISPLAY 'YOU MAY ENTER "TERSE" TO SET TERSE MODE.'
002810 DISPLAY ' '
002820 DISPLAY 'OR "VERBOSE" TO SET VERBOSE MODE.'
002830 DISPLAY ' '
002840 DISPLAY 'YOU MAY ENTER "DELETE" TO DELETE A FACILITY.'
002850 DISPLAY ' '
002860 DISPLAY 'YOU MAY ENTER "HELP" FOR A COMPLETE EXPLANATION '
002870 'OF THE SYSTEM'
002880 DISPLAY ' '
002890 DISPLAY 'YOU MAY ENTER "LIST" FOR A COMPLETE DISPLAY '
002900 'OF A SINGLE AIRPORT WITH ALL CODES '
002910 ' (TRANSLATED) '
002920 DISPLAY 'YOU MAY ENTER "STOP" TO END A SESSION.'
002930 DISPLAY ' '
002940 DISPLAY ' '
002950 INTRO-SYSTEM-EXIT. EXIT.
002960 EJECT
002970 SET-VERBOSE.
002980 DISPLAY USER-ID ' DO YOU WANT VERBOSE OR TERSE '
002990 ' COMMUNICATIONS? (Y/N) '
003000 DISPLAY ' '
003010 SET-VERBOSE=01.
003020 ACCEPT YES=NO.
003030 IF QUESTION
003040 DISPLAY '"VERBOSE" COMMUNICATIONS ARE MORE '
003050 ' DETAILED THAN "TERSE" '

```

```

003060      GO TO SET-VERBOSITY.
003070      DISPLAY ' '.
003080      IF YES-NO = 'V'
003090          MOVE 'V' TO COMM-MODE
003100          DISPLAY 'VERBOSE MODE SET'
003110          DISPLAY ' '
003120          GO TO SET-VERBOSITY-EXIT.
003130      IF YES-NO = 'T'
003140          MOVE 'T' TO COMM-MODE
003150          DISPLAY 'TERSE MODE SET'
003160          DISPLAY ' '
003170          GO TO SET-VERBOSITY-EXIT.
003180          DISPLAY 'INVALID RESPONSE - ' YES-NO ' - REENTER'
003190          GO TO SET-VERBOSITY-01.
003200      SET-VERBOSITY-EXIT. EXIT.
003210      EJECT
003220      OPEN-RTN.
003230          OPEN I-Q-DATABASE.
003240          OPEN INPUT DCOD-FILE.
003250          OPEN-RTN-EXIT. EXIT.
003260      SPECIAL-OPEN.
003270          DISPLAY 'DO YOU WISH TO "LOAD" THE DATABASE?'.
003280          ACCEPT YES-NO
003290          IF YES NEXT SENTENCE ELSE GO TO SPECIAL-OPEN-EXIT.
003300          DISPLAY 'FORMATTING DATABASE'.
003310          OPEN OUTPUT DATABASE.
003320          DISPLAY 'FORMATTED'.
003330          PERFORM LOAD-RAN-DATA THRU LOAD-RAN-OATA-EXIT.
003340          CLOSE DATABASE.
003350          CLOSE DCOD-FILE.
003360          MOVE OUTPUT-COUNT TO COUNT-MASK.
003370          DISPLAY 'DATABASE FORMATTED AND LOADED WITH ' COUNT-MASK
003380          ' RECORDS'.
003390          STOP RUN.
003400      SPECIAL-OPEN-EXIT. EXIT.
003410      EJECT
003420      TERMINATE-RTN.
003430          CLOSE DATABASE.
003440          DISPLAY ' '.
003450          DISPLAY 'SESSION SUMMARY'.
003460          DISPLAY ' '.
003470          DISPLAY ' '.
003480          MOVE ADD-COUNT TO COUNT-MASK.
003490          DISPLAY COUNT-MASK ' RECORDS ADDED'.
003500          MOVE UPDATE-COUNT TO COUNT-MASK.
003510          DISPLAY COUNT-MASK ' RECORDS UPDATED'.
003520          MOVE OUTPUT-COUNT TO COUNT-MASK.
003530          IF OUTPUT-COUNT > 0
003540              DISPLAY COUNT-MASK ' NEW SURVEY RECORDS LOADED INTO '
003550              'DATABASE'.
003560      TERMINATE-RTN-EXIT. EXIT.
003570      EJECT
003580      GET-REGION.
003590          DISPLAY ' '.
003600          IF TERSE DISPLAY 'REGION?'
003610              ELSE DISPLAY 'ENTER REGION (OR COMMAND)'.
003620      GR-01.
003630          ACCEPT IN-KEY.
003640          PERFORM CHECK-FOR-COMMANDS THRU CHECK-FOR-COMMANDS-EXIT.
003650          IF COMMAND GO TO GET-REGION.
003660          PERFORM CHECK-REG THRU CHECK-REG-EXIT.

```

```

003670 IF INVALID-REGION DISPLAY M1 'BAD REGION' M2
003680 GO TO GR-01.
003690 GET-REGION-EXIT.
003700 EXIT.
003710 GET-LOCID.
003720 DISPLAY ' '.
003730 IF TERSE DISPLAY 'LOCID?'
003740 ELSE DISPLAY 'ENTER LOCID ' .
003750 GR-01.
003760 ACCEPT IN-LOCID.
003770 IF IN-LOCID = 'EXIT' OR 'END'
003780 DISPLAY 'LOCID ENDED WITH "EXIT" OR "END" COMMAND'
003790 DISPLAY 'PROCESSING WILL BEGIN WITH NEXT REGION'
003800 GO TO PROCESS-DATA.
003810 PERFORM CHECK-LOCID THRU CHECK-LOCID-EXIT.
003820 IF INVALID-LOCID DISPLAY M1 'BAD LOCID' M2 GO TO GL-01.
003830 GET-LOCID-EXIT. EXIT.
003840 EJECT.
003850 PROCESS-DATA.
003860 PERFORM GET-REGION THRU GET-REGION-EXIT.
003870 MOVE ' ' TO DISPLAY-SLASH.
003880 PERFORM GET-LOCID THRU GET-LOCID-EXIT.
003890 MOVE EDIT-REGION TO KEY-REGION.
003900 MOVE EDIT-LOCID ID KEY-LOCID.
003910 PERFORM COMPUTE-KEY THRU COMPUTE-KEY-EXIT.
003920 MOVE '0' TO INVALID-KEY-SW.
003930 READ DATABASE INVALID KEY MOVE '1' TO INVALID-KEY-SW.
003940 IF NOT INVALID-KEY AND DELETE-FLAG = '1'
003950 DISPLAY ' *** WARNING - FACILITY ' IN-KEY
003960 ' HAS BEEN PREVIOUSLY DELETED'
003970 DISPLAY 'HOWEVER, YOU MAY STILL ELECT TO ADD THE RECORD'
003980 DISPLAY ' '
003990 PERFORM ADD-NEW-RECORD-01 THRU ADD-NEW-RECORD-EXIT
004000 GO TO PROCESS-DATA-EXIT.
004010 IF INVALID-KEY
004020 PERFORM ADD-NEW-RECORD THRU ADD-NEW-RECORD-EXIT ELSE
004030 PERFORM UPDATE-RECORD THRU UPDATE-RECORD-EXIT.
004040 PROCESS-DATA-EXIT. EXIT.
004050 EJECT
004060 ADD-NEW-RECORD.
004070 DISPLAY ' '
004080 IF TERSE DISPLAY 'ADD?' ELSE
004090 DISPLAY 'FACILITY ' IN-KEY
004100 ' DOES NOT CURRENTLY EXIST'
004110 DISPLAY 'WOULD YOU LIKE TO ADD THE RECORD? (YES/NO?)'.
004120 ADD-NEW-RECORD-01.
004130 ACCEPT YES-NO.
004140 IF QUESTION DISPLAY 'THE FACILITY ' IN-KEY
004150 ' HAS NOT BEEN PREVIOUSLY ADDED TO THE SYSTEM'
004160 DISPLAY 'YOU MAY ELECT TO DO SO NOW BY ENTERING "YES"'
004170 GO TO ADD-NEW-RECORD.
004180 IF NOPE GO TO ADD-NEW-RECORD-EXIT.
004190 IF YES NEXT SENTENCE ELSE
004200 DISPLAY M1 YES-NO M2
004210 GO TO ADD-NEW-RECORD.
004220 MOVE 'ADD' TO UP-TYPE.
004230 PERFORM ADD-A-RECORD THRU ADD-A-RECORD-EXIT.
004240 ADD-NEW-RECORD-EXIT. EXIT.
004250 EJECT
004260 ADD-A-RECORD.
004270 PERFORM INIT-RECORD THRU INIT-RECORD-EXIT.

```



```

004280 PERFORM ADD-FIELDS THRU ADD-FIELDS-EXIT.
004290 ADD-END.
004300 PERFORM WRITE-ROUTINE THRU WRITE-ROUTINE-EXIT.
004310 DISPLAY 'RECORD ADDED'.
004320 DISPLAY ' '.
004330 ADD 1 TO ADD-COUNT.
004340 ADD-A-RECORD-LXIT. EXIT.
004350 INIT-RECORD.
004360 MOVE SPACES TO DATA-REC.
004370 MOVE RECORD-KEY TO DATA-REC-KEY.
004380 INIT-RECORD-EXIT. EXIT.
004390 EJECT.
004400 ADD-FIELDS.
004410 MOVE '0' TO END-SW.
004420 MOVE 1 TO USE-SUB.
004430 PERFORM ADD-USE-CODES THRU ADD-USE-CODES-EXIT.
004440 MOVE '1' TO END-SW.
004450 PERFORM ADD-USE-CODES THRU ADD-USE-CODES-EXIT.
004460 VARYING USE-SUB FROM 2 BY 1 UNTIL USE-SUB > 40
004470 OR FINISHED-SW = '1'.
004480 IF USE-SUB > 40 DISPLAY 'THAT IS ALL OF THE USE CODES'.
004490 ' I HAVE ROOM FOR'.
004500 ADD-FIELDS-EXIT. EXIT.
004510 EJECT.
004520 UPDATE-RECORD.
004530 IF TERSE DISPLAY 'UPDATE?' ELSE
004540 DISPLAY 'FACILITY' IN-KEY
004550 'ALREADY EXISTS'.
004560 DISPLAY 'WOULD YOU LIKE TO UPDATE THE RECORD?'.
004570 'YES/NO?'.
004580 ACCEPT YES-NO.
004590 IF QUESTION
004600 DISPLAY 'FACILITY' IN-KEY
004610 'HAS ALREADY BEEN ADDED TO THE SYSTEM'.
004620 DISPLAY 'YOU MAY UPDATE USE CODES WITHIN THIS'.
004630 'RECORD BY ENTERING "YES"'.
004640 GO TO UPDATE-RECORD.
004650 IF NOPE OR YES-NO = SPACES GO TO UPDATE-RECORD-EXIT.
004660 IF YES NEXT SENTENCE ELSE
004670 DISPLAY M1 YES-NO M2
004680 GO TO UPDATE-RECORD.
004690 MOVE 'UPD' TO OP-TYPE.
004700 MOVE '1' TO END-SW.
004710 GET-FIELD-NUMBER.
004720 IF VERBOSE DISPLAY 'ENTER THE USE CODE YOU WISH'.
004730 'TO UPDATE (USE CODE)', 'END', '2' ELSE
004740 DISPLAY 'USE CODE?'.
004750 ACCEPT IN-USE-CODE.
004760 GET-FIELD-NUMBER-01.
004770 IF IN-USE-CODE = '?'
004780 DISPLAY 'AT THIS POINT I NEED TO KNOW WHICH'.
004790 'USE CODE YOU WISH TO CHANGE'.
004800 DISPLAY 'YOU TELL ME THIS BY ENTERING THE USE CODE'.
004810 GO TO GET-FIELD-NUMBER.
004820 IF IN-USE-CODE = 'END'
004830 DISPLAY 'UPDATE COMPLETED'.
004840 GO TO UPDATE-RECORD-EXIT.
004850 IF IN-USE-CODE = 'EXIT'
004860 DISPLAY 'UPDATE ENDED DUE TO "EXIT" COMMAND'.
004870 GO TO UPDATE-RECORD-EXIT.
004880 UPDATE-THE-FIELD.

```



```

004890 PERFORM UPDATE-USES THRU UPDATE-USES-EXIT.
004900 END-UPDATE.
004910 PERFORM WRITE-ROUTINE THRU WRITE-ROUTINE-EXIT.
004920 DISPLAY 'RECORD UPDATED'.
004930 DISPLAY ' '.
004940 ADD 1 TO UPDATE-COUNT.
004950 END-UPDATE-01.
004960 IF TERSE DISPLAY 'NEXT USE CODE OK 'END' ELSE
004970 DISPLAY 'ENTER TIME NEXT USE CODE OR 'END''.
004980 ACCEPT IN-USE-CODE.
004990 IF IN-USE-CODE = '1'.
005000 DISPLAY 'YOU MAY NOW ENTER ADDITIONAL USE CODES'.
005010 DISPLAY 'THAT YOU WOULD LIKE TO UPDATE IN FACILITY'.
005020 IN-KEY.
005030 GO TO END-UPDATE-01.
005040 GO TO GET-FIELD-NUMBER-01.
005050 UPDATE-RECORD-EXIT. EXIT.
005060 EJECT.
005070 CHECK-FOR-COMMANDS.
005080 MOVE 'Q' TO COMMAND-FOUND.
005090 IF IN-KEY = 'TERSE'.
005100 MOVE '1' TO COMMAND-FOUND.
005110 MOVE '1' TO COMMAND-FOUND.
005120 DISPLAY 'TERSE MODE SET'.
005130 GO TO CHECK-FOR-COMMANDS-EXIT.
005140 IF IN-KEY = 'VERBOSE'.
005150 MOVE '1' TO COMMAND-FOUND.
005160 MOVE '1' TO COMMAND-FOUND.
005170 DISPLAY 'VERBOSE MODE SET'.
005180 GO TO CHECK-FOR-COMMANDS-EXIT.
005190 IF IN-KEY = 'DELETE'.
005200 PERFORM DELETE-RTN THRU DELETE-RTN-EXIT.
005210 MOVE '1' TO COMMAND-FOUND.
005220 GO TO CHECK-FOR-COMMANDS-EXIT.
005230 IF IN-KEY = 'HELP'.
005240 PERFORM INTRO-SYSTEM THRU INTRO-SYSTEM-EXIT.
005250 MOVE '1' TO COMMAND-FOUND.
005260 GO TO CHECK-FOR-COMMANDS-EXIT.
005270 IF IN-KEY = 'LIST'.
005280 PERFORM LIST-RTN THRU LIST-RTN-EXIT.
005290 MOVE '1' TO COMMAND-FOUND.
005300 GO TO CHECK-FOR-COMMANDS-EXIT.
005310 IF IN-KEY = 'STOP'.
005320 MOVE '1' TO MORE-UPDATES.
005330 GO TO PROCESS-DATA-EXIT.
005340 IF IN-KEY = 'EXIT'.
005350 MOVE '1' TO COMMAND-FOUND.
005360 DISPLAY 'EXIT IS INVALID AT THIS POINT'.
005370 DISPLAY 'PERHAPS YOU MEANT "STOP"'.
005380 GO TO CHECK-FOR-COMMANDS-EXIT.
005390 CHECK-FOR-COMMANDS-EXIT. EXIT.
005400 EJECT.
005410 WRITE-ROUTINE.
005420 PERFORM SORT-THE-CODES THRU SORT-THE-CODES-EXIT.
005430 IF OP-TYPE = 'UPD'.
005440 PERFORM UPD-WRITE THRU UPD-WRITE-EXIT.
005450 GO TO WRITE-ROUTINE-EXIT.
005460 MOVE 'Q' TO INVALID-KEY-SW.
005470 REWRITE DATA-REC INVALID KEY.
005480 MOVE '1' TO INVALID-KEY-SW.
005490 IF INVALID-KEY-SW = '1'.

```

```

005500 MOVE '0' TO INVALID-KEY-SW
005510 WRITE DATA-REC INVALID KEY
005520 MOVE '1' TO INVALID-KEY-SW.
005530 IF INVALID-KEY-SW = '1'
005540 DISPLAY 'UNABLE TO ADD FACILITY', IN-KEY
005550 PERFORM SEVERE-ERROR THRU SEVERE-ERROR-EXIT.
005560 WRITE-ROUTINE-EXIT. EXIT.
005570 UPD-WRITE.
005580 MOVE '0' TO INVALID-KEY-SW.
005590 PERFORM COMPUTE-KEY THRU COMPUTE-KEY-EXIT.
005600 KWRITE DATA-REC INVALID KEY
005610 MOVE '1' TO INVALID-KEY-SW.
005620 IF INVALID-KEY-SW = '1'
005630 DISPLAY 'UNABLE TO UPDATE FACILITY', IN-KEY
005640 PERFORM SEVERE-ERROR THRU SEVERE-ERROR-EXIT.
005650 UPD-WRITE-EXIT. EXIT.
005660 COMPUTE-KEY.
005670 COMPUTE TALLY = BIN1 + BIN2.
005680 DIVIDE 17 INTO TALLY GIVING TALLY REMAINDER TRACK-ID.
005690 IF TRACK-ID < 0 COMPUTE TRACK-ID = TRACK-ID + 1.
005700 COMPUTE-KEY-EXIT. EXIT.
005710 SEVERE-ERROR.
005720 DISPLAY 'CONTACT TRANS SYSTEMS CORP (202) 281-1500 - '
005730 'BEFORE USING SYSTEM AGAIN'.
005740 PERFORM TERMINATE-RTN THRU TERMINATE-RTN-EXIT.
005750 STOP RUN.
005760 SEVERE-ERROR-EXIT. EXIT.
005770 EJECT
005780 GET-FIELD-VALUE.
005790 MOVE '0' TO END-FLAG. EXIT-FLAG.
005800 MOVE 10 TO TALLY.
005810 PERFORM ACCEPT-SCREEN THRU ACCEPT-SCREEN-EXIT.
005820 IF SHORT-BUFFER = 'END' MOVE '1' TO END-FLAG
005830 PERFORM END-EXIT-ANALYZER THRU END-EXIT-ANALYZER-EXIT
005840 GO TO GET-FIELD-VALUE.
005850 IF SHORT-BUFFER = 'EXIT' MOVE '1' TO EXIT-FLAG
005860 PERFORM END-EXIT-ANALYZER THRU END-EXIT-ANALYZER-EXIT
005870 GO TO GET-FIELD-VALUE.
005880 PERFORM END-LENGTH THRU END-LENGTH-EXIT
005890 VARYING TALLY FROM TALLY BY -1
005900 UNTIL CHAR (TALLY) NOT = ' ' OR
005910 TALLY = 0.
005920 IF TALLY > FIELD-LENGTH
005930 DISPLAY M1 'FIELD TOO LONG' M2
005940 GO TO GET-FIELD-VALUE.
005950 DISPLAY ' '.
005960 GET-FIELD-VALUE-EXIT. EXIT.
005970 FIND-LENGTH.
005980 FIND-LENGTH-EXIT. EXIT.
005990 ACCEPT-SCREEN.
006000 ACCEPT-SCREEN-EXIT. EXIT.
006010 ACCEPT-SCREEN-BUFFER.
006020 END-EXIT-ANALYZER.
006030 IF OP-TYPE = 'UPD' AND END-FLAG = '1'
006040 DISPLAY 'END' TREATED LIKE 'EXIT' IN UPDATE MODE.
006050 DISPLAY 'RECORD NOT UPDATED'
006060 GO TO END-UPDATE-01.
006070 IF OP-TYPE = 'UPD' AND EXIT-FLAG = '1'
006080 DISPLAY 'RECORD NOT UPDATED'
006090 GO TO END-UPDATE-01.
006100 IF OP-TYPE = 'ADD' AND END-FLAG = '1' AND END-SW = '0'

```

```

006110 DISPLAY 'YOU MAY NOT "END" THE ADD OPERATION UNTIL'
006120 ' AFTER ENTERING AT LEAST ONE USE CODE - REENTER'
006130 GO TO END-EXIT-ANALYZER-EXIT
006140 IF OP-TYPE = 'ADD' AND END-FLAG = '1' AND END-SW = '1'
006150 DISPLAY 'RECORD COMPLETED WITH "END" COMMAND'
006160 GO TO ADD-END
006170 IF OP-TYPE = 'ADD' AND EXIT-FLAG = '1'
006180 DISPLAY 'RECORD NOT ADDED DUE TO "EXIT" COMMAND'
006190 GO TO ADD-NEW-RECORD-EXIT
006200 END-EXIT-ANALYZER-EXIT. EXIT
006210 DELETE-RTN
006220 IF TERSE-DELETE-DELETE-REGION? ELSE
006230 DISPLAY 'ENTER THE REGION OF THE RECORD YOU WISH TO'
006240 'DELETE'
006250 PERFORM GR-01 THRU GET-REGION-EXIT
006260 PERFORM GET-LOCID THRU GET-LOCID-EXIT
006270 MOVE EDIT-REGION TO KEY-REGION
006280 MOVE EDIT-LOCID TO KEY-LOCID
006290 PERFORM COMPUTE-KEY THRU COMPUTE-KEY-EXIT
006300 READ DATABASE INVALID KEY
006310 DISPLAY 'FACILITY - IN-KEY - DOES NOT EXIST'
006320 ' - DELETE COMMAND ENDED'
006330 GO TO DELETE-RTN-EXIT
006340 IF DELETE-FLAG = '1'
006350 DISPLAY 'FACILITY - IN-KEY - PREVIOUSLY DELETED'
006360 ' - DELETE COMMAND ENDED'
006370 GO TO DELETE-RTN-EXIT
006380 MOVE '1' TO DELETE-FLAG
006390 REWRITE DATA-REC INVALID KEY
006400 DISPLAY 'UNABLE TO DELETE RECORD'
006410 ' - CONTACT TRANS SYSTEMS CORP (202) 281-1500'
006420 GO TO DELETE-RTN-EXIT
006430 DISPLAY 'FACILITY - IN-KEY - DELETED'
006440 DELETE-RTN-EXIT. EXIT
006450 EJECT
006460 LIST-RTN
006470 IF TERSE-DELETE-DELETE-REGION? ELSE
006480 DISPLAY 'ENTER THE REGION OF THE RECORD YOU WANT LISTED'
006490 PERFORM GR-01 THRU GET-REGION-EXIT
006500 PERFORM GET-LOCID THRU GET-LOCID-EXIT
006510 MOVE EDIT-REGION TO KEY-REGION
006520 MOVE EDIT-LOCID TO KEY-LOCID
006530 MOVE '0' TO INVALID-KEY-SW
006540 PERFORM COMPUTE-KEY THRU COMPUTE-KEY-EXIT
006550 READ DATABASE INVALID KEY
006560 MOVE '1' TO INVALID-KEY-SW
006570 IF INVALID-KEY
006580 DISPLAY 'FACILITY - IN-KEY'
006590 ' DOES NOT EXIST - LIST COMMAND ENDED'
006600 GO TO LIST-RTN-EXIT
006610 PERFORM LIST-RECORD THRU LIST-RECORD-EXIT
006620 LIST-RTN-EXIT. EXIT
006630 EJECT
006640 LOAD-RAM-DATA
006650 OPEN INPUT RAW-DATA
006660 LOAD-RAM-01
006670 READ-TYPE-1
006680 MOVE SPACES TO RECORD-THRU-AREA
006690 READ RAW-DATA INTO ARRAY-RECORD (1) AT END GO TO LOAD-DUNE
006700 ADD 1 TO RAM-INPUT-COUNT
006710 READ-TYPE-2

```

B-17

```

006720 IF EOF-SW = 1 GO TO LOAD-DONE.
006730 MOVE SPACES TO ARRAY-RECORD (2).
006740 READ RAM-DATA INTO ARRAY-RECORD (2)
006750 AT END MOVE '1' TO EOF-SW
006760 PERFORM MOVER THRU MOVER-EXIT
006770 GO TO LOAD-DONE.
006780 ADD 1 TO RAM-INPUT-COUNT.
006790 PERFORM MUVER THRU MOVER-EXIT.
006800 LOAD-DONE.
006810 DISPLAY 'RAW INPUT RECORDS = ' RAM-INPUT-COUNT.
006820 DISPLAY 'OUTPUT RECORDS WRITTEN = ' OUTPUT-COUNT.
006830 DISPLAY 'OUTPUT USE CODES = ' CODE-COUNT.
006840 CLOSE RAM-DATA.
006850 LOAD-RAM-DATA-EXIT. EXIT.
006860 MOVER.
006870 MOVE SPACES TO DATA-REC.
006880 MOVE ARRAY-REGION (1) TO DATA-REC-REGION KEY-REGION.
006890 MOVE ARRAY-LOCID (1) TO DATA-REC-LOCID KEY-LOCID.
006900 MOVE 1 TO REC-SUB OUT-SUB IN-SUB.
006910 MOVER-FIRST-RECORD.
006920 PERFORM MOVE-THE-CODES THRU MOVE-THE-CODES-EXIT.
006930 IF ARRAY-REGION (1) NOT = ARRAY-REGION (2) OR
006940 ARRAY-LOCID (1) NOT = ARRAY-LOCID (2) MOVE
006950 ARRAY-RECORD (2) TO ARRAY-RECORD (1)
006960 PERFORM WRITE-RAM-DATA THRU WRITE-RAM-DATA-EXIT
006970 GO TO READ-TYPE-2.
006980 MOVER-SECOND-RECORD.
006990 MOVE 2 TO REC-SUB.
007000 MOVE 1 TO IN-SUB.
007010 PERFORM MOVE-THE-CODES THRU MOVE-THE-CODES-EXIT.
007020 PERFORM WRITE-RAM-DATA THRU WRITE-RAM-DATA-EXIT.
007030 GO TO READ-TYPE-1.
007040 MOVER-EXIT.
007050 WRITE-RAM-DATA.
007060 MOVE '0' TO INVALID-KEY-SW.
007070 PERFORM COMPUTE-KEY THRU COMPUTE-KEY-EXIT.
007080 WRITE DATA-REC INVALID KEY MOVE '1' TO INVALID-KEY-SW.
007090 IF INVALID-KEY
007100 DISPLAY '*** AN ERROR HAS OCCURRED AFTER ' OUTPUT-COUNT
007110 ' RECORDS'.
007120 ADD 1 TO OUTPUT-COUNT.
007130 WRITE-RAM-DATA-EXIT. EXIT.
007140 MOVE-THE-CODES.
007150 IF ARRAY-USE-CODE (REC-SUB, IN-SUB) = ' '
007160 GO TO MOVE-THE-CODES-EXIT.
007170 MOVE ARRAY-USE-CODE (REC-SUB, IN-SUB) TO USE-CODE-ARRAY
007180 (OUT-SUB).
007190 ADD 1 TO CODE-COUNT.
007200 ADD 1 TO OUT-SUB IN-SUB.
007210 IF IN-SUB > 18 GO TO MOVE-THE-CODES-EXIT
007220 ELSE GO TO MOVE-THE-CODES.
007230 MOVE-THE-CODES-EXIT. EXIT.
007240 EJECT
007250 UPDATE-USERS.
007260 PERFORM CHECK-USE-CODE THRU CHECK-USE-CODE-EXIT.
007270 IF INVALID-USE-CODE
007280 DISPLAY '1' 'INVALID USE CODE' M2
007290 GO TO END-UPDATE-01.
007300 PERFORM LOCATE-USE-CODE THRU LOCATE-USE-CODE-EXIT.
007310 IF LOCATE-SUB > 40
007320 PERFORM INSERT-NEW-CODE THRU INSERT-NEW-CODE-EXIT

```



```

007330 ELSE PERFORM DELETE-CHANGE-ROUTINE THRU
007340 DELETE-CHANGE-ROUTINE-EXIT.
007350 UPDATE-USE-EXIT. EXIT.
007360 INSERT-NEW-CODE.
007370 IF TERSE DISPLAY 'USE CODE' IN-USE-CODE
007380 ELSE DISPLAY 'USE CODE' IN-USE-CODE
007390 , DOES NOT EXIST.
007400 DISPLAY 'WOULD YOU LIKE TO ADD IT? (YES/NO/EXIT)'.
007410 MOVE 4 TO FIELD-LENGTH.
007420 PERFORM GET-FIELD-VALUE THRU GET-FIELD-VALUE-EXIT.
007430 IF SHORT-BUFFER = 'YES' OR 'Y' GO TO INSERT-CONTINUE.
007440 IF SHORT-BUFFER = 'NO' OR 'N' GO TO END-UPDATE-01.
007450 INSERT-CONTINUE.
007460 PERFORM FIND-NEXT-CELL THRU FIND-NEXT-CELL-EXIT.
007470 MOVE IN-USE-CODE TO USE-CODE-ARRAY (LOCATE-SUB).
007480 DISPLAY 'USE CODE' IN-USE-CODE 'INSERTED (' LOCATE-SUB ')'.
007490 INSERT-NEW-CODE-EXIT. EXIT.
007500 DELETE-CHANGE-ROUTINE.
007510 IF TERSE DISPLAY 'REPLACEMENT CODE?'
007520 ELSE DISPLAY 'ENTER THE REPLACEMENT USE CODE'
007530 , 'CARRIER "RETURN" FOR DELETE'.
007540 MOVE 4 TO FIELD-LENGTH.
007550 PERFORM GET-FIELD-VALUE THRU GET-FIELD-VALUE-EXIT.
007560 IF SHORT-BUFFER = SPACES
007570 MOVE SPACES TO USE-CODE-ARRAY (LOCATE-SUB)
007580 DISPLAY 'USE CODE' IN-USE-CODE 'DELETED'
007590 GO TO DELETE-CHANGE-ROUTINE-EXIT.
007600 MUST-BE-CHANGE.
007610 MOVE IN-USE-CODE TO OLD-USE-CODE.
007620 MOVE SHORT-BUFFER TO IN-USE-CODE.
007630 PERFORM CHECK-USE-CODE THRU CHECK-USE-CODE-EXIT.
007640 IF INVALID-USE-CODE
007650 DISPLAY M1 'INVALID USE CODE' M2
007660 ELSE GO TO POST-CHANGE.
007670 MOVE 4 TO FIELD-LENGTH.
007680 PERFORM GET-FIELD-VALUE THRU GET-FIELD-VALUE-EXIT.
007690 GO TO MUST-BE-CHANGE.
007700 POST-CHANGE.
007710 MOVE IN-USE-CODE TO USE-CODE-ARRAY (LOCATE-SUB).
007720 DISPLAY 'USE CODE' OLD-USE-CODE 'CHANGED TO '
007730 USE-CODE-ARRAY (LOCATE-SUB).
007740 DELETE-CHANGE-ROUTINE-EXIT. EXIT.
007750 ADD-USE-CODES.
007760 MOVE '0' TO FINISHED-SW.
007770 ADD-USE-01.
007780 IF TERSE DISPLAY 'CODE?' ELSE
007790 DISPLAY 'ENTER USE CODE ("CR" OR "END" WHEN FINISHED)'.
007800 ADD-USE-02.
007810 MOVE 4 TO FIELD-LENGTH.
007820 PERFORM GET-FIELD-VALUE THRU GET-FIELD-VALUE-EXIT.
007830 IF SHORT-BUFFER = SPACES AND USE-SUB = 1
007840 DISPLAY M1 'YOU MUST ENTER AT LEAST ONE VALID USE '
007850 'CODE'.
007860 GO TO ADD-USE-02.
007870 IF SHORT-BUFFER = SPACES
007880 DISPLAY 'RECORD COMPLETED WITH CARRIER RETURN (CR)'
007890 MOVE '1' TO FINISHED-SW
007900 GO TO ADD-USE-CODES-EXIT.
007910 MOVE SHORT-BUFFER TO IN-USE-CODE.
007920 PERFORM CHECK-USE-CODE THRU CHECK-USE-CODE-EXIT.
007930 IF INVALID-USE-CODE DISPLAY M1 'BAD USE CODE' M2

```



```

007940 GO TO ADD-USE-02.
007950 MOVE SHORT-BUFFER TO USE-CODE-ARRAY (USE-SUB).
007960 ADD-USE-CODES-EXIT, EXIT.
007970 LIST-RECORD.
007980 LIST-RECORD-LOOP.
007990 MOVE 1 TO LIST-SUB.
008000 LIST-REC-01.
008010 MOVE USE-CODE-ARRAY (LIST-SUB) TO UCDE.
008020 MOVE KEY2 TO NON-KEY REC-KEY.
008030 READ DCOD-FILE INVALID KEY.
008040 DISPLAY 'UNABLE TO DECODE USE CODE ' KEY2
008050 LIST COMMAND ENDED.
008060 GO TO LIST-RECORD-EXIT.
008070 IF LIST-SUB = 1
008080 DISPLAY 'REGION: ' DATA-REC-REGION
008090 LOCID: ' DATA-REC-LOCID
008100 DISPLAY ' '
008110 DISPLAY ' ' UCDE ' ' USERCODE.
008120 ADD 1 TO LIST-SUB.
008130 IF LIST-SUB > 40 OR USE-CODE-ARRAY (LIST-SUB) =
008140 SPACES NEXT SENTENCE ELSE
008150 GO TO LIST-REC-01.
008160 LIST-RECORD-EXIT, EXIT.
008170 CHECK-REG.
008180 IF IN-REG NOT = 'AAL' AND 'ACE' AND 'AEA' AND 'AGL'
008190 AND 'ANE' AND 'ANN' AND 'ARM' AND 'ASO' AND 'ASM'
008200 AND 'ANE'
008210 MOVE '1' TO REGION-OK, ELSE MOVE '0' TO REGION-OK.
008220 CHECK-REG-EXIT, EXIT.
008230 CHECK-LOCID.
008240 MOVE EDIT-LOCID TO LOC.
008250 MOVE KEY1 TO REC-KEY NON-KEY.
008260 MOVE '0' TO INVALID-KEY-SW LOCID-OK.
008270 READ DCOD-FILE INVALID KEY MOVE '1' TO INVALID-KEY-SW.
008280 IF INVALID-KEY-SW = '1' MOVE '1' TO LOCID-OK.
008290 CHECK-LOCID-EXIT, EXIT.
008300 LOCATE-USE-CODE.
008310 MOVE 1 TO LOCATE-SUB.
008320 LOCATE-01.
008330 IF IN-USE-CODE = USE-CODE-ARRAY (LOCATE-SUB)
008340 GO TO LOCATE-SUB.
008350 ADD 1 TO LOCATE-SUB.
008360 IF LOCATE-SUB < 41 GO TO LOCATE-01.
008370 LOCATE-USE-CODE-EXIT, EXIT.
008380 CHECK-USE-CODE.
008390 MOVE '0' TO CODE-OK.
008400 MOVE IN-USE-CODE TO UCDE.
008410 MOVE KEY2 TO NON-KEY REC-KEY.
008420 MOVE '0' TO INVALID-KEY-SW.
008430 READ DCOD-FILE INVALID KEY MOVE '1' TO INVALID-KEY-SW.
008440 IF INVALID-KEY-SW = '1' MOVE '1' TO CODE-OK.
008450 CHECK-USE-CODE-EXIT, EXIT.
008460 FIND-NEXT-CELL.
008470 MOVE 1 TO LOCATE-SUB.
008480 FN-01.
008490 IF USE-CODE-ARRAY (LOCATE-SUB) = ' '
008500 GO TO FIND-NEXT-CELL-EXIT.
008510 ADD 1 TO LOCATE-SUB.
008520 IF LOCATE-SUB > 40
008530 DISPLAY 'THERE IS NO MORE ROOM TO ADD NEW USE CODES '
008540 'IN FACILITY ' IN-KEY

```

```

008550 DISPLAY 'CONTACT TRANS SYSTEMS COMP - 281-1500'
008560 GO TO FIND-NEXT-CELL-EXIT.
008570 GO TO FN-01.
008580 FIND-NEXT-CELL-EXIT. EXIT.
008590 SORT-THE-CODES.
008600 SORT SORT-FILE
008610 UN ASCENDING KEY SORT-USE-CODE
008620 INPUT PROCEDURE IS BEFORE-SORT
008630 OUTPUT PROCEDURE IS AFTER-SORT.
008640 IF SORT-RETURN NUT = ZERO
008650 DISPLAY '*** UNSUCCESSFUL SORT'
008660 DISPLAY 'SOME USE CODES MAY BE OUT OF SEQUENCE'
008670 DISPLAY 'HOWEVER, NO DATA HAS BEEN LOST'
008680 DISPLAY 'CONTACT TRANS SYSTEMS CORPORATION - 281-1500'
008690 PERFORM TERMINATE-RTN THRU TERMINATE-NIN-EXIT
008700 STOP RUN.
008710 MOVE HOLD-CODE-TABLE TO USES.
008720 SORT-THE-CODES-EXIT. EXIT.
008730 BEFORE-SORT SECTION.
008740 MOVE SPACES TO HOLD-CODE-TABLE.
008750 MOVE 1 TO SORT-SUB.
008760 BS-01.
008770 IF USE-CODE-ARRAY (SORT-SUB) NOT = SPACES
008780 MOVE USE-CODE-ARRAY (SORT-SUB) TO SORT-USE-CODE
008790 RELEASE SORT-REC.
008800 ADD 1 TO SORT-SUB.
008810 IF SORT-SUB < 41 GO TO BS-01.
008820 BEFORE-SORT-EXIT. EXIT.
008830 AFTER-SORT SECTION.
008840 MOVE J TO SORT-SUB.
008850 AS-01.
008860 RETURN SORT-FILE AT END GO TO AFTER-SORT-EXIT.
008870 IF DUP-USE-CODE = SORT-USE-CODE
008880 DISPLAY 'DUPLICATE USE CODE OF ' SORT-USE-CODE
008890 ' FOUND AND DISCARDED' ELSE
008900 MOVE SORT-USE-CODE TO HOLD-CODE-ARRAY
008910 (SORT-SUB) DUP-USE-CODE
008920 ADD 1 TO SORT-SUB.
008930 IF SORT-SUB > 40 DISPLAY 'LOGIC ERROR - RECORD OVERFLOW'
008940 DISPLAY 'CONTACT TRANS SYSTEMS - 281-1500'.
008950 GO TO AS-01.
008960 AFTER-SORT-EXIT. EXIT.

```

```

4. EDB.COBOL(USES)
00010 IDENTIFICATION DIVISION.
00020 PROGRAM-ID.
00030 'USES'.
00040 AUTHOR.
00050 ROBERT STANLEY.
00060 REMARKS.
00070 ENVIRONMENT DIVISION.
00080 CONFIGURATION SECTION.
00090 SOURCE-COMPUTER. IBM-370.
00100 OBJECT-COMPUTER. IBM-370.
00110 SPECIAL-NAMES.
00120 C01 IS TO-NEXT-PAGE.
00130 INPUT-OUTPUT SECTION.
00140 FILE-CONTROL.
00150 SELECT USE-IN ASSIGN TO UT-S-USEIN.
00160 SELECT USE-PRT ASSIGN TO UT-S-USEPRT.
00230 DATA DIVISION.
00240 FILE SECTION.
00250 FD USE-IN
00260 RECORDING MODE IS F
00270 BLOCK CONTAINS 0 RECORDS
00280 RECORD CONTAINS 153 CHARACTERS
00290 LABEL RECORDS ARE STANDARD
00300 DATA RECORD IS USEIN.
00310 01 USEIN PIC X(153).
00430 FD USE-PRT
00440 BLOCK CONTAINS 0 RECORDS
00450 RECORD CONTAINS 133 CHARACTERS
00460 RECORDING MODE IS F
00470 LABEL RECORDS ARE STANDARD
00480 DATA RECORD IS PRT-REC.
00490 01 PRT-REC PIC X(133).
00730 WORKING-STORAGE SECTION.
00740 77 AP-HLD PIC X(42) VALUE SPACES.
00760 77 ERR-SW PIC 9 VALUE ZEROS.
00761 01 PRT-IT SYNC.
00762 05 FILLER PIC XXX.
00763 05 PRTUSE PIC XXXX.
00764 05 FILLER PIC XXX.
00765 05 PRT-USE PIC X(96).
00766 05 FILLER PIC X(27).

```

```

00820 01 ASTR-LINE SYNC.
00830 05 FILLER
00840 01 PRT1 SYNC.
00850 05 FILLER
00860 05 FILLER
00870 05 FILLER
00880 05 PRT1-APNAME
00890 05 FILLER
00900 05 PRT2 SYNC.
00910 05 FILLER
00920 05 REGPT
00930 05 FILLER
00940 05 LOCPT
00950 05 FILLER
00960 05 CDEPT
00970 05 FILLER
00980 05 TEXTPT
00990 05 FILLER
01000 05 APNAMEPT
01010 01 WS-ACCUMMS SYNC.
01020 05 IN-CNT
01030 05 PRT-CNT
01040 01350 PROCEDURE DIVISION.
01050 OPEN-FILES.
01060 OPEN INPUT USE-IN
01070 OUTPUT USE-PRT.
01080 MOVE SPACES TO PRT2.
01090 READ-TRANS.
01100 READ USE-IN INTO PRT2 AT END GO TO EOJ.
01110 ADD 1 TO IN-CNT.
01120 IF APNAMEPT = AP-HLD
01130 NEXT SENTENCE
01140 ELSE
01150 WRITE PRT-REC FROM ASTR-LINE AFTER ADVANCING 2 LINES
01160 MOVE APNAMEPT TO PRT1-APNAME AP-HLD
01170 WRITE PRT-REC FROM PRT1 AFTER ADVANCING 2 LINES
01180 MOVE 1 TO ERR-SW.
01190 MOVE SPACES TO PRT-IT.
01200 MOVE TEXTPT TO PRT-USE.
01210 IF ERR-SW = 1
01220 MOVE 0 TO ERR-SW
01230 WRITE PRT-REC FROM PRT-IT AFTER ADVANCING 2 LINES
01240 ELSE
01250 WRITE PRT-REC FROM PRT-IT AFTER ADVANCING 1 LINES.
01260 ADD 1 TO PRT-CNT.
01270 GO TO READ-TRANS.
01280 EOJ.
01290 DISPLAY 'RECS READ IN: ' IN-CNT.
01300 DISPLAY 'RECS PRINTED: ' PRT-CNT.
01310 CLOSE USE-IN USE-PRT.
01320 STOP RUN.

```

Job Control Language and TSO Command Lists for USE8, SORTER, LIST and COBOL

```

1.  USE8 CLIST
00010 BEGIN: WRITENR DO YOU WANT TO RUN BATCH OR INTERACTIVE (B/I)?...
00020 READ $RUNTYPE
00030 IF $LENGTH($RUNTYPE)=0 THEN GOTO BEGIN
00040 IF $RUNTYPE = BATCH THEN SET $RUNTYPE=B
00050 IF $RUNTYPE = INTERACTIVE THEN SET $RUNTYPE=I
00060 IF $RUNTYPE = B THEN GOTO BATCH
00070 WRITE I WILL HAVE TO SORT THE DATABASE INTO THE PROPER SEQUENCE
00080 WRITE PRIOR TO RUNNING - THIS MONT TAKE LONG
00090 EX SORTER
00100 IF $LASTCC EQ 0 THEN GOTO INTERACT
00110 WRITE THE SORT HAS FAILED - CONTACT PROJECT MANAGER
00120 GOTO EQJ
00130 INTERACT: FREEALL
00140 TERM LINESIZE(133)
00150 AL DA(*) F(SYSOUT)
00160 AL DA(*) F(USEPRT)
00170 AL DA('FAA110.MAST.DECODE.AUG78.DATA') F(DCOD) VOL(AVPFAA)
00180 ALLOC F(USEOUT) DUMMY DA('FAA118.AIRPORT.MAR78.USE.EASYTREV.DATA')
00190 ALLOC F(USEIN) DA('FAA130.EDB.AUG78.USE8.DATA')
00200 CALL 'FAA130.EDB.LOAD(USE8)'
00210 END
00220 BATCH: E 'FAA130.USE8.CNTL' CNTL
00230 V OFF
00240 B1:
00250 READ $COP
00260 IF $LENGTH($COP)=0 THEN GOTO B1
00270 WRITE WHAT PRIORITY WOULD YOU LIKE (1 - 15)?
00280 WRITE (THE HIGHER THE PRIORITY THE FASTER THE JOB WILL RUN)
00290 WRITE (HOWEVER, IT WILL COST MORE)
00300 READ $PRI
00310 IF $LENGTH($PRI)=0 THEN SET $PRI=5
00320 IF $PRI = 15 THEN SET $PRI=14
00330 TOP
00340 C 10 /PP/$PRI/
00350 TOP
00360 C 10 300 /CPS/$COP/
00370 TOP
00380 BATCHEND: S USE8.HOLD.CNTL
00390 WRITE I HAVE JUST CREATED AND SAVED A JOBSTREAM TO EXECUTE THE USE8 PROGRAM
00400 WRITE I WILL NOW SORT THE DATABASE INTO THE PROPER ORDER FOR USE8
00410 WRITE DONT TOUCH THE TERMINAL - THIS MONT TAKE LONG
00420 END
00430 EX 'FAA130.SORTER.CLIST'
00440 IF $LASTCC NE 0 THEN GOTO BADSORT
00450 WRITE THE SORT HAS COMPLETED - I CAN NOW SUBMIT YOUR BATCH RUN OF USE8
00460 SUB USE8.HOLD.CNTL
00470 BADSORT:DEL USE8.HOLD.CNTL
00480 EQU:WRITE END OF USE8 PROCESSING

```



```

2.  USE8.CNTL
    00010 //FAA1308 JOB (ED,EQ110,000000,A90),NOTIFY=FAA130,PTY=PP,TIME=(1)
    00020 //USE8 EXEC PGM=USE8,PARM='BATCH'
    00030 //STEPLIB DD DSN=FAA130.EDB.LOAD,DISP=SHR
    00040 //SYSOUT DD SYSOUT=A
    00050 //USEPRD DD SYSOUT=(A,,61) COPIES=CPS
    00070 //DCOD DD DSN=FAA110.MAST.DECODE.AUG78.DATA,DISP=SHR
    00090 //USEOUT DD DSN=FAA118.AIRPORT.MAR78.USE.EASYTREV.DATA,DISP=SHR
    00100 //USEIN DD DSN=FAA130.EDB.AUG78.USE8.DATA,DISP=SHR

3.  COBOL.CLIST
    00010 FREEALL
    00030 ATTR OBJLIST BLKSIZE(80) LRECL(80) DSORG(PS) RECFM(F)
    00040 WRITER ENTER PROGRAM NAME ...
    00050 READ $PGM
    00060 ALLOC F(SYSLIN) DA($PGM..OBJ) NEW CATALOG SPACE(20,10) TRACKS USING(OBJLIST)
    00070 COBOL $PGM
    00080 AL F(SYSLIB) DA('SYS1.COBLIB')
    00090 LINK $PGM LOAD(EDB.LOAD($PGM)) COBLIB
    00100 DEL $PGM..OBJ

4.  LIST.CLIST
    00010 BEGIN:WRITER WHAT DATASET DO YOU WANT LISTED?...
    00020 READ $DSN
    00030 IF $LENGTH($DSN)=0 THEN GOTO BEGIN
    00040 WRITER HOW MANY COPIES DO YOU WANT?...
    00050 READ $COPIES
    00060 IF $LENGTH($COPIES)=0 THEN SET $COPIES=1
    00070 WRITER WHAT FORM TYPE (WHITE OR GREEN)?...
    00080 READ $FORM
    00090 IF $LENGTH($FORM)=0 THEN SET $FORM=A,,51
    00100 IF $FORM=WHITE THEN SET $FORM=A,,61
    00110 IF $FORM NE A,,61 THEN SET $FORM=A,,51
    00120 E 'FAA130.GENER.CNTL' CNTL
    00130 V OFF
    00140 TOP
    00150 C 10 200 /USERDSN/FAA130.$DSN/
    00160 TOP
    00170 C 10 200 /CPY/$COPIES/
    00180 TOP
    00190 C 10 200 /A,,51/$FORM/
    00200 TOP
    00230 SUB *
    00240 END
    00250 WRITER MORE LISTINGS?...
    00300 READ $ANS
    00310 IF $ANS=YES THEN GOTO BEGIN

```

5. UPDATE.CLIST
 00010 FREEALL
 00020 TERM LINESIZE(80)
 00030 ATTR BDAMLIST BLKSIZE(167) LRECL(167) OPTCD(E) LIMCT(4) DSORG(DA) RECFM(F)
 00040 ALLOC DA(*) DDN(SYSOUT)
 00050 ALLOC DA(*) DDN(SYSIN)
 00051 ALLOC DUMMY DDN(SORTMSG)
 00060 ALLOC DA('FAA130.EDB.AUG78.USE8.DATA') DDN(RAWINPUT)
 00070 ALLOC DA('FAA130.EDB.DATABASE') DDN(DATABASE) USING(BDAMLIST) SHR -
 00080 SPACE(25) TRACKS
 00081 ALLOC DA('FAA110.MAST.DECODE.AUG78.DATA') F(DCOD) SHR
 00090 CALL 'FAA130.EDB.LOAD(UPDATE)'
 00100 FREEALL

6. SORTER.CLIST
 00010 FREEALL
 00020 TERM LINESIZE(133)
 00021 AL DA('FAA130.EDB.DATABASE') F(DATABASE) SHR
 00030 AL DA(*) F(SYSOUT)
 00031 AL F(SORTWK01) SPACE(20,20) TRACKS NEW DELETE
 00032 AL F(SORTWK02) SPACE(20,20) TRACKS NEW DELETE
 00033 AL F(SORTWK03) SPACE(20,20) TRACKS NEW DELETE
 00034 AL F(SORTLIB) DA('SYS1.SORTLIB') SHR
 00035 AL F(SYSIN) DUMMY
 00036 AL F(SORTMSG) DUMMY
 00050 ALLOC F(RAWINPUT) DA('FAA130.EDB.AUG78.USE8.DATA')
 00060 CALL 'FAA130.EDB.LOAD(SORTER)'